DNSトラフィック解析及び負荷改善に関する研究

DNS トラフィック解析及び負荷改善に関する研究

２０１５年　３月

藤原　和典

DNS トラフィック解析及び負荷改善に関する研究

藤原　和典

システム情報工学研究科

筑波大学

２０１５年　３月

論文概要

ドメイン名システム (Domain Name System, DNS) は，人間にとり使い易いドメイン名を機械が扱いやすい IP アドレスなどに変換する等，インターネットの利便性にとり欠く事のできないシステムである．

DNS には，権威 DNS サーバ，フルリゾルバ，スタブリゾルバの三つの構成要素がある．権威 DNS サーバはドメイン名に対応する委任情報や IP アドレスなどの情報を保持し，フルリゾルバからのクエリに応答する．権威 DNS サーバは各ドメインごとに運用し，ルートから TLD への委任情報を提供するルート DNS サーバ，TLD からユーザ組織ドメイン名への委任情報を提供する TLD DNS サーバ，ユーザ組織の情報を提供する各組織の権威 DNS サーバなどが存在する．スタブリゾルバはエンドノードの名前解決ライブラリに相当し，ユーザからのクエリをフルリゾルバに送る．フルリゾルバはルート DNS サーバの情報を事前に保持しており，スタブリゾルバからの検索要求をもとに，ルート DNS サーバから順に名前ツリーをたどり，名前解決を行い，結果をスタブリゾルバに返す．フルリゾルバでは名前解決の効率化のために検索途中の情報や検索結果を指定された時間キャッシュすることができる．

すべての名前解決は，フルリゾルバによってルート DNS サーバから始まるため，フルリゾルバとルート DNS サーバの働きは特に重要である．さらに，現在のインターネットを維持するためには，ルート DNS サーバへのクエリ量を適切な範囲に収めることが重要である．

今日では IPv6 の普及によるタイプ AAAA のクエリの増加や，Web ブラウザでの閲覧待ち時間短縮のための DNS プリフェッチの実装，コンテンツデリバリネットワーク (CDN) での広域負荷分散のための小さな TTL 値の

設定などの影響で，DNS への負荷は増大してきている．また，DNSSEC の普及や，DNS に公開鍵証明書をのせる技術 (DANE/TLSA) の標準化により，今後さらに DNS への負荷が増大することが見込まれ，その負荷低減は重要な研究テーマである．

このような背景に基づき，本研究では，(1) 中規模のフルリゾルバのクエリ分析，(2) キャッシュ効率増大や遅延低減のための適切な設定の提案，(3) ルート DNS サーバのクエリ分析及びフルリゾルバのクエリ分析，(4) ルート DNS サーバへのクエリ低減方法の提案を行なう．提案手法の有効性を大学のフルリゾルバのデータをもとにしたシミュレーションにより，検証した．各提案の概要を以下に示す．

(1) 中規模のフルリゾルバのクエリ分析

DNS の動作を分析するため，フルリゾルバのクエリ分析は古くから行われている．本研究では，まず近年の IPv6 や CDN の影響を評価するため，中規模なフルリゾルバとして筑波大学情報環境機構学術情報メディアセンターの運用するフルリゾルバの入出力パケットを分析した．そのフルリゾルバは全学のユーザを対象としており，学生寮や Wi-Fi 接続の通常の PC が参照するため，利用帯域等のデータから中規模 ISP のフルリゾルバに近いと考えられる．時期によるが一ヶ月で 8800 から 11000 のクライアント IP アドレスを観測した．

キャプチャデータの分析の結果，IPv6 対応 OS の普及により IPv6 アドレス (AAAA) クエリが増加したため，クライアントからのクエリが 41%増加したこと，CDN などの DNS を使用した広域負荷分散サービスの普及により，小さな TTL 値，CNAME，外部名ネームサーバホスト名の使用が増えていることがわかった．結果として，エラー応答の割合の増加やキャッシュヒット率の低下，応答遅延の増大がみられた．

また，2012 年 6 月 6 日の World IPv6 Launch の前後で AAAA クエリに存在応答が戻る割合が 2.0%から 21.2%に増加した．これは，World IPv6 Launch 以降，よく使用されるドメイン名の IPv6 対応が進んだことを示している．

(2) キャッシュ効率増大や遅延低減のための適切な設定の提案

(1) の結果は，(a) 無用な AAAA クエリの低減，(b) 適切に大きな TTL 値の設定，(c)CNAME の削減，(d) 内部名ネームサーバホスト名の適切な設定，が重要な運用課題である事を示している．これらの結果を背景に，本研究では以下の指摘と提案を行った．

(a)IPv6 対応 OS(比較的新しいバージョンの Microsoft Windows や Apple Mac OS X，Linux など) は IPv6 接続性がない場合でも AAAA クエリをフルリゾルバに送信する．これら IPv6 対応 OS が，IPv6 接続性の不存在を検知し，不要な AAAA クエリを省略することで，フルリゾルバの負荷を軽減することができる．

(b)DNS を利用した広域負荷分散で使用される小さな TTL 値は，キャッシュヒット率を下げ，フルリゾルバの負荷を上げ，権威 DNS サーバへのクエリ数を増大させる．名前解決にかかる平均時間も，TTL 値 300 以下で 31.7 ミリ秒が，TTL 値が 300 を超える場合には 25.1 ミリ秒と，TTL 値が大きい方が短い．適切に大きな TTL 値を使用する事で名前解決に要する時間を短縮できる．

(c)CNAME を使用する複雑なドメイン名の名前解決に要する平均時間は 30.4 ミリ秒であったが，CNAME を使用しないドメイン名の場合は 24.9 ミリ秒であった．CNAME の使用を最小限にすることで名前解決に要する時間を短縮できる．

(d) 外部名ネームサーバホスト名を使用するドメイン名の名前解決に要する時間は平均 33.1 ミリ秒であったが，すべて内部名ネームサーバホスト名を使用するドメイン名の場合は平均 22.5 ミリ秒であり，内部名ネームサーバホスト名を適切に使用することで名前解決の時間を短縮できる．

これらの施策を行なうことで，フルリゾルバの負荷を低減でき，名前解決に要する時間を短縮できる．

(3) ルート DNS サーバ及び筑波大学のフルリゾルバのクエリ分析

ルート DNS サーバでは年に一度パケットキャプチャを実施し，研究者にデータを提供している．そのデータを分析し，多くの IP アドレスから大量の本来不必要なクエリがルート DNS サーバに送られ，余分な負荷が発生している事を示した．具体的には，3 万以上の IP アドレスが 48 時間に 10 万以上のクエリをルート DNS サーバに送っていた．筑波大学のフルリゾルバも本来不要なクエリを大量にルートに送っていたことがわかり，動作を確認したところ，一般的な設定がされた中規模以上のフルリゾルバ共通の振舞いである事がわかった．現象を詳細に解析した結果，世界で広く使われている BIND 9 はルート DNS サーバに無駄なクエリを送ること，存在しない TLD に対するクエリによるルート DNS サーバへのクエリも多いことがわかった．

(4) ルート DNS サーバへのクエリ低減方法の提案

ルート DNS サーバへの不要クエリが多い問題に対し，本研究では (a) フルリゾルバの設定による改善法，(b) フルリゾルバの名前解決とキャッシュアルゴリズムの改善法，(c)NSEC リソースレコードを使った改善法，の 3 つの手法を示し，それぞれの効果を定量的に評

価した．

(a) フルリゾルバの設定による改善法は，BIND 9 とは異なる Unbound という実装を用い，キャッシュサイズを増やすことである．

(b) アルゴリズムの改善では，フルリゾルバでのデータの扱いを改善した理想的なフルリゾルバアルゴリズムを規定し，名前解決のシミュレーションを行ない，ルート DNS サーバへのクエリ数が (a) と同様となることを示した．この改善には RFC 2181 の変更を必要とする．

(c)NSEC リソースレコードを使った改善法は DNSSEC の不存在応答検証の仕組みを用い，ルート DNS サーバに到達する存在しない TLD のクエリを減らす手法であり，Unbound にパッチをあてて実装し，評価を行なった．結果として (a)(b) では削減できない存在しない名前についてのルート DNS サーバへのクエリを 78%減らすことができた．この改善には RFC 4035 の小規模な変更を必要とする．

(a)(c) の組み合わせで，ルート DNS サーバへのクエリを 1/29 に減らすことができる．ルート DNS サーバへのクエリを減らすことで，ユーザからのクエリの応答時間の短縮とフルリゾルバの負荷の低減も同時に達成される．

以上の評価と提案により，ユーザからのクエリへの応答時間の短縮，フルリゾルバの負荷の低減，ルート DNS サーバへのクエリの低減が可能となる．

目 次

図 目 次

表 目 次

第1章　序論

1.1　研究の背景

ドメイン名システム (Domain Name System, DNS) [46][47][48] は，人間にとり使い易いドメイン名を機械が扱いやすい IP アドレスなどに変換する等，インターネットの利便性にとり欠く事のできないシステムである．

DNS には，権威 DNS サーバ，フルリゾルバ，スタブリゾルバの三つの構成要素がある．権威 DNS サーバはドメイン名に対応する委任情報や IP アドレスなどの情報を保持し，フルリゾルバからのクエリに応答する．権威 DNS サーバは各ドメインごとに運用し，ルートから TLD への委任情報を提供するルート DNS サーバ，TLD からユーザ組織ドメイン名への委任情報を提供する TLD DNS サーバ，ユーザ組織の情報を提供する各組織の権威 DNS サーバなどが存在する．

すべての名前解決は，フルリゾルバによってルート DNS サーバから始まるため，フルリゾルバとルート DNS サーバの働きは特に重要である．さらに，現在のインターネットを維持するためには，ルート DNS サーバへのクエリ量を適切な範囲に収めることが重要である．

今日では IPv6 の普及によるタイプ AAAA のクエリの増加や，Web ブラウザでの閲覧待ち時間短縮のための DNS プリフェッチの実装，コンテンツデリバリネットワーク (CDN) での広域負荷分散のための小さな TTL 値の設定などの影響で，DNS への負荷は増大してきている．また，DNSSEC の普及や，DNS に公開鍵証明書をのせる技術 (DANE/TLSA) の標準化により，今後さらに DNS への負荷が増大することが見込まれ，その負荷低

減は重要な研究テーマである．

1.2 研究の目的と意義

本研究は，このような背景を考慮し，現在のインターネットでの DNS への負荷の状態を把握し，DNS への負荷の低減方法を検討することを目的とした．特に IPv6 の普及とコンテンツデリバリネットワークなどの大規模配信事業者の使用する DNS 設定がフルリゾルバに及ぼす影響を評価した．ユーザからの名前解決要求を受けとるフルリゾルバの負荷を低減する方法を把握することは，ユーザの名前解決遅延時間を低減することにもつながる．また，ルート DNS サーバへのクエリ数を削減することで，将来に渡って現在の DNS を用いたインターネットが持続できる技術についても検討した．

1.3 本論文の構成

本論文の構成を以下に示す．

まず，第 2 章では，DNS 技術の概要を紹介したあと，インターネットでのトラフィック解析，DNS トラフィック及びフルリゾルバのアルゴリズムについての過去の研究を概観し，DNS に関する課題を明確化する．

第 3 章では，IPv6 対応 OS の普及や CDN などの大規模配信事業者による DNS を用いた負荷分散の影響を調べるために筑波大学のフルリゾルバのクエリを分析し，キャッシュ効率増大や遅延低減のための適切な設定の提案を行なう．

第 4 章では，ルート DNS サーバ及び及び筑波大学のフルリゾルバのクエリ分析を行ない，ルート DNS サーバに非常に多くの本来不必要なクエリを送る IP アドレスが多数存在し，一般的な設定の中規模以上のフルリゾルバがその原因の一つであることを示す．次にルート DNS サーバへの

クエリ数の削減策を提案し評価を行なう．提案手法の有効性を筑波大学のフルリゾルバのデータをもとにしたシミュレーションにより，検証する．

最後に第5章では本研究で得られた結果の総括を行なう．

第2章　DNSの概要と関連研究

2.1　緒言

第1章で述べたように，インターネットの名前解決を担当するDNSへの負荷が増大しつつあり[16][24] [26]，DNSのトラフィック解析の重要性が増している．(jp TLDへのクエリ数増大については，図3.5に示す．)

本章では，DNSの概要を紹介したあと，DNS及びインターネットでのトラフィック解析についての過去の研究を概観し，DNSに関する課題を明確化する．

2.2　DNSの概要

ドメイン名はルートを起点とし，63文字以下のラベルをドットで連結したもので，ラベルごとに階層をなすドメイン名空間を構成する．ドメイン名空間は，ルートを根とし，ルートの子ノードにラベルが一つのトップレベルドメイン名 (Top Level Domain name, TLD)，TLDの子ノードに一般組織のドメイン名などが存在する木構造である．DNSでは，ドメイン名の階層ごとに階層以下の管理主体を変更することができ，それを委任と呼ぶ．委任された単位をゾーンと呼ぶ．主な委任点は，ルート，TLD，実際の利用者が利用する一般組織のドメイン名などである．委任する側を親 (ゾーン) と呼び，委任される側を子 (ゾーン) と呼ぶ．

DNSはドメイン名，タイプ，クラスをキーとするデータベースであり，タイプごとに決められた形式の値を保持する．キーと値の組をリソースレ

コード (RR) と呼ぶ．キーのうちドメイン名をリソースレコードの所有者名 (Owner name) と呼ぶ．また，一つのキーに対して異なる値を持つ複数の RR を設定でき，それらの集合をリソースレコードセット (RRSet) と呼ぶ．タイプには，IPv4 アドレスを示す A や，IPv6 アドレスを示す AAAA，委任情報を示す NS，別名変換のための正式名を保持する CNAME，ゾーンの管理情報を保持する SOA などがある．それぞれのタイプのリソースレコードを A リソースレコード (A RR) や AAAA リソースレコード (AAAA RR) と呼ぶ．インターネットではクラスとして IN が用いられる．

DNS には，権威 DNS サーバ，フルリゾルバ，スタブリゾルバの三つの構成要素がある．権威 DNS サーバは親ゾーンの DNS サーバから委任されたゾーンの情報を保持し，その範囲のドメイン名空間の管理権限を持つ．権威 DNS サーバはフルリゾルバからのクエリに応答し，管理権限を持つ情報と委任情報を提供する役目を持つ．各ゾーンごとに権威 DNS サーバを運用し，ルート DNS サーバ，TLD DNS サーバ，一般組織の権威 DNS サーバなどが存在する．ルート DNS サーバは TLD への委任情報を保持する．ドメイン名登録者は TLD 運用組織 (レジストリ) にドメイン名と委任情報を登録する．TLD DNS サーバは登録者のドメイン名への委任情報を保持する．

委任情報は，委任対象のゾーン名，タイプ NS に委任先のネームサーバホスト名を値として持たせた RRSet である．委任先のネームサーバホスト名が，委任するゾーン名の子孫のドメイン名であるか，委任を含むゾーン名の子孫のドメイン名の場合，そのネームサーバホスト名を内部名であるとし，それ以外を外部名であるという．内部名のネームサーバホスト名への委任の場合，ネームサーバホスト名の A RR あるいは AAAA RR を委任情報として追加することができ，それをグルーと呼ぶ．権威 DNS サーバは委任情報を応答する時，グルーを添付する．フルリゾルバは名前解決時に外部名ネームサーバホスト名を含む委任情報を受け取ると，内

部名ネームサーバホスト名の場合に比べ，余分なネームサーバホスト名の IPv4，IPv6 アドレスの解決を行なう必要がある．

例えば example.com ゾーンのネームサーバホスト名が内部名である ns.example.com の場合，委任情報として ns.example.com の A RR と AAAA RR を委任情報に追加できる．具体的には， ns.example.com の IPv4 アドレスが 192.2.0.1 の場合，com ゾーン内に以下の情報が記述される．

example.com. IN NS ns.example.com.

ns.example.com. IN A 192.2.0.1

example.com ゾーンのネームサーバホスト名が外部名の ns.example.jp の場合，com の子孫ではないため，グルーを追加できず，

example.com. IN NS ns.example.jp.

という情報が登録される．この例では，example.com の名前解決を要求されたフルリゾルバは外部名の委任情報を受け取ったところで example.com の名前解決を中断し， ns.example.jp の名前解決を行なう必要がある．

一般のドメイン名登録者は自ドメイン名のゾーンの情報を管理し，そのなかに自組織のホスト名から IP アドレスへの対応や，メールサーバなどの情報を登録する．また，ゾーン内に，親ゾーンに登録した委任情報と同じ NS RR，ネームサーバホストの情報 (A RR，AAAA RR) とゾーンの管理情報 (SOA RR) を権威ある情報として記述する必要がある．

スタブリゾルバはエンドノードの名前解決ライブラリに相当し，ユーザからのクエリをフルリゾルバに送る．フルリゾルバはルート DNS サーバの情報を事前に保持しており，スタブリゾルバからの検索要求をもとに，ルート DNS サーバから順にドメイン名空間の木構造を探索し，名前解決を行い，結果をスタブリゾルバに返す．フルリゾルバでは名前解決の効率化のために検索途中の情報や検索結果を指定された時間キャッシュすることができる．権威 DNS サーバからの応答に含まれる RRSet は Time

to live (TTL) 値を含み，キャッシュ可能な時間 (秒) を示す．

すべての名前解決は，フルリゾルバによってルート DNS サーバから始まるため，フルリゾルバとルート DNS サーバの働きは特に重要である．さらに，現在のインターネットを維持するためには，ルート DNS サーバへのクエリ量を適切な範囲に収めることが重要である．

また，DNS の信頼性を増すために DNS に公開鍵暗号を用いた署名検証機能を追加する DNS Security Extensions (DNSSEC) [4, 6, 5, 39] が開発され，実装が進んだ．2010 年にルートが DNSSEC に対応し，前後して多くの TLD が DNSSEC に対応した．ドメイン名登録者のドメイン名が DNSSEC に対応し，フルリゾルバに DNSSEC 検証機能を追加して有効化すると DNS の応答が正しいことを検知可能になる．一方，DNSSEC では公開鍵や署名情報の追加によるトラフィックの増大や，フルリゾルバでの署名検証による負荷増大が懸念されている．

DNSSEC では，タイプとして DNSKEY，RRSIG，DS，NSEC，NSEC3 が追加された．DNSKEY RR は，ゾーンの公開鍵を保持する．DS RR は，委任先の DNSSEC 署名鍵のハッシュを委任元で保持する．RRSIG RR は RRSet の署名を保持する．NSEC RR と NSEC3 RR は，名前が存在する範囲を示すとともに，名前に存在するタイプを列挙することで，名前が存在しない範囲を示し，不存在証明に使用される．

2.3 DNS に関する研究事例

DNS に関する研究はインターネットの初期からおこなわれている．

Danzig らは 1992 年にルート DNS サーバや複数の著明なサイトの権威 DNS サーバへのクエリを収集し，異常な DNS トラフィックの増加とその原因を示し，対策を提案した [20]．欠陥のある権威 DNS サーバとフルリゾルバを修正すれば広域の DNS トラフィックは減少すること，例えば，す

べてのフルリゾルバが，BIND 4 で実装されているネガティブキャッシュを実装することを提案した．

その後も継続して研究が行なわれているため，それぞれの研究を権威 DNS サーバ，スタブリゾルバ，フルリゾルバ，DNS への攻撃の分析，及び DNS プロトコルの改善に分類して紹介する．

2.3.1 権威 DNS サーバに関する研究事例

DNS サーバへのトラフィックとそれによる負荷は DNS サーバの増強計画を立てるための重要な情報となるため，ルート DNS サーバ運用者は研究者と連係し，ルート DNS サーバのトラフィックを分析してきた．The Cooperative Association for Internet Data Analysis (CAIDA) は 2001 年にルート DNS サーバのうちの 1 システム，f.root-servers.net へのクエリを 18 日間収集し，分析を行なった [12]．その後，ルート DNS サーバオペレータと CAIDA は，ルート DNS サーバなどで 2006 年 1 月から年に 1 度 48 時間のデータ収集を行なう活動，“A Day in the Life of the Internet” (DITL) を開始した [17]．

現在では，データ収集活動と研究者への解析環境の提供を Domain Name System Operations Analysis and Research Center (DNS-OARC) が提供している．DITL への参加は強制ではないため，データ収集に参加するルート DNS サーバは一部であり，複数のサーバを用いて分散運用している場合にすべてを収集できているとは限らず，集まったデータがすべてである．また，ルート DNS サーバへのクエリ量は週や日付の影響を受けるが，DITL データセットは一年のうちの 48 時間のデータであるため，経年の変化を分析するには注意が必要であるが，ルート DNS サーバへのクエリ傾向を知る唯一の情報源である．

Liu らは最初の結果を “Two days in the life of the DNS anycast root

servers” [43] として報告した．CAIDA は定期的に分析を行なっており，Castro らはルート DNS サーバの情報を解析し，ルート DNS サーバから見た DNS の負荷増大について報告した [15][16]．その中で，いくつかのルート DNS サーバについて，2007 年，2008 年，2009 年のクエリ量を比較している．c.root-servers.net では 2 年で 50%増加，k.root-servers.net では 2 年で 33%増加，m.root-servers.net では 2 年で 17%増加している．また，ルート DNS サーバへのクエリの 98%が不要なものであると分析した．

筆者は DNS-OARC のルートデータセットを調査し，2006 年から 2014 年のデータを単純に比較し，単純に集計した [24]．2008 年と 2014 年には 8 個のルート DNS サーバのデータを集計でき，2011 年と 2013 年が 10 個，2012 年が 9 個のため，それらの年の収集できたクエリ数を比較すると，2008 年に 110 億クエリ，2011 年，2012 年に 260 億クエリ，2014 年に 321 億クエリであり，ルート DNS サーバへのクエリは 6 年間で 3 倍に増加している．

また，Larson らは顕著に多くのクエリをルート DNS サーバや com，net などの TLD DNS サーバに送るフルリゾルバの振舞いについて報告し，実装に関する提案を行なった [38]．

Brownlee らは，大学のネットワークからルート DNS サーバ及び TLD DNS サーバに送られるクエリとその応答を観測することで，大学から各 DNS サーバへの遅延の変化を定量的に評価した [13][14]．

Lee らは 2003 年にルート DNS サーバの配置の最適化の提案を行なった [40]．

Larson らは 2011 年 3 月から 2012 年 6 月までの com, net TLD の権威 DNS サーバのクエリ情報を収集し，分析結果を運用者のミーティングである RIPE 68 DNS ワーキンググループで発表した [59]．主な結果は，com, net の名前解決を試みる IPv4 アドレスは全 IPv4 アドレスの 2%であること，多くのフルリゾルバはタイプ A とタイプ MX 両方の名前解決を試み

ること，AAAA クエリも増加していること，上位 100 クライアントがクエリの 10%を，上位 5000 クライアントが 50%を，上位 20 万クライアントがクエリの 90%を生成していること，RD ビットがセットされたクエリの多くはタイプ MX を問い合わせていることを示した．

Sharma らは 2014 年に移動体の通信に適した名前解決の仕組みの提案と比較を行なった [51]．彼らは，次世代の名前解決のデザイン，試作，評価を行ない，その中で DNS との比較を行ない，DNS を用いた仕組みは提案手法よりも劣るという結論を示した．彼らの枠組では，DNS を用いて移動体の情報を扱う時に，TTL 値として 0 に近い小さな値を用い，DNS への負荷が大きい仮定となっている．

2.3.2 スタブリゾルバに関する研究事例

Broido らは 2006 年にエンドノードからのホスト名登録の試みである DNS Update [55] を送るアドレスについて，RFC 1918[50] で指定されたプライベートアドレスの権威 DNS サーバへのクエリ情報を解析することで評価し，その権威 DNS サーバには毎時数百万以上の無駄な DNS Update パケットが到達していること，原因の 99%以上は Microsoft Windows 2000，Windows XP の標準設定の問題であることを示した [11]．プライベートアドレスの DNS サーバは AS112[1] という仕組みで運用されている．

Iinou らは 2010 年に DNS-OARC ワークショップにてスタブリゾルバからフルリゾルバへのクエリ数増大について報告した [26]． 主な結論は，顧客からのクエリは 1 年で 50%増加したことと，増加の主な原因は Firefox の DNS プリフェッチで，DNS クエリを 2 倍にしている事である．

2.3.3　フルリゾルバに関する研究事例

フルリゾルバに関する研究事例も多い．Jung らは 2002 年にフルリゾルバの動作を解析し，DNS パフォーマンスとキャッシュ効率について報告した [30]．彼らは二つの大学で一週間の DNS のトラフィックを収集・分析し，クライアントの視点からみた権威 DNS サーバの動作と，DNS パフォーマンス及びフルリゾルバのキャッシュ効率について報告した [30]．そのなかで，フルリゾルバから権威 DNS サーバに送られるクエリのうち 23%が無応答，13%がその他のエラーで，ルート DNS サーバに送られるクエリの 27%がエラーとなるものであることを示した．また，彼らは TTL 値と DNS のキャッシュ効率の関係について報告し，A リソースレコードの TTL 値を数百 (秒) 程度に小さくしてもキャッシュヒット率の低下は限定的であること，10〜20 程度のクライアントで DNS のキャッシュを共有しても効果は少ないことを示し，CDN などにより広まった 200〜300(秒) という小さな TTL 値の使用は，広域ネットワークでの DNS 関連のトラフィックを大規模には増やしていないことを示した．

Wessels らは，2004 年に複数の大学からの DNS トラフィックを分析し，フルリゾルバの実装上の問題により，ルート DNS サーバや TLD DNS サーバに大量の無駄クエリが送られていることを示し，BIND 9 はルートに無駄なクエリを送る傾向があることや，到達できない権威 DNS サーバがある場合には BIND 8 などの古い実装には問題があり，到達できないアドレスに大量のクエリを送る傾向があることを示した [58]．

Internet Research Task Force(IRTF) の Routing Research Group(RRG) では識別子 (Identifier) と位置識別情報 (Locator) の分離などのルーティング技術の研究を行なっており，DNS Resource Records for the Identifier-Locator Network Protocol (ILNP) を RFC 6742[9] として実験のために標準化した．ILNP ではノード識別子情報や，位置識別情報を DNS のリソー

スレコードとして保持し，機器が移動するとDynamic Update[55] プロトコルを使用し，位置情報を書き換えるという提案である．移動する機器の情報登録に使う場合にはキャッシュの影響を考慮する必要があり，権威DNSサーバを変更しても，DNSのフルリゾルバはTTL値によって指定された時間のあいだ情報をキャッシュするため，TTL値を小さくする必要がある．そこで，Bhattiらは，2011年のIEEE Global Internet Symposiumにて，大学の学科の権威DNSサーバのTTL値を小さくした場合のフルリゾルバ及び権威DNSサーバへの負荷について評価し，TTL値を0にしてもクエリは2倍にしかならないことを示した[10]．しかし，彼らの実験ではクエリ頻度が平均2.36クエリ/秒，最大68クエリ/秒と低く，大規模に検索される場合を評価していない．

神屋らは2009年にDNSの持つキャッシュ機能がクライアントへ与える影響を調査した[32]．特にTTL値を0にしてもキャッシュの効果を無効にできない場合があることを示した．

Bernhardらは2010年に公開フルリゾルバサービスがCDNなどのDNSを使用した広域負荷分散に悪影響があることを示した[2]．具体的には，公開フルリゾルバサービスと，ローカルISPのフルリゾルバの応答の違いを比較し，大規模配信事業者を使っているドメイン名の場合には公開フルリゾルバサービスの応答とエンドユーザが使用しているISPのフルリゾルバの応答が異なり，ISPのフルリゾルバの応答のほうがエンドユーザに近いサーバを返した．

Kocらは2011年のDNS EASYワークショップにてDNSのグローバル参照モデルを提案した．彼らはエンドユーザのクエリ傾向を分析し，よく使用されているDNSサーバソフトウェアの振舞の分析結果に基づき，DNSモデルを作成した[62]．

Jiangらは2012年にRFC 2181[22]のランキングについてのフルリゾルバの実装上の問題点を指摘し，過去に存在したドメイン名が消えない幽

霊ドメイン名が存在しうることを示した [29].

Yu らは 2012 年によく使用されているフルリゾルバを評価し，複数の DNS サーバが設定されているゾーンでのサーバ選択ルールや，無応答サーバの判定などの実装上の間違いの指摘を行ない，無応答サーバに大量のクエリを送る実装があることを示した [63].

Gao らは Farsight Security, Inc. の SIE (Security Information Exchange) からデータの提供を受け，2012/12/9〜2012/12/22 の二週間，US 東西と EU の 600 個所以上の地理的に分散したフルリゾルバから 260 億以上の DNS クエリと応答を収集し，AAAA クエリが増加していること，存在しない TLD に関するクエリが増加していること，同じ内容を連続して問い合わせるクエリが多いことを示し，分析結果を元にマルウェアの検知方法を提案した [25]．SIE ではプライバシの観点からユーザのクエリを見ず，フルリゾルバと権威サーバ間のクエリをフルリゾルバ側で収集し，Farsight Security のサーバにデータを収集している．

Lian らは 2013 年の USENIX Security Symposium にて，インターネット広告を使って 50 万以上の地理的に分散したクライアントからの名前解決を評価し，DNSSEC の影響を評価した結果を報告した [41]．主な結論は，比較的小数のユーザのみが DNSSEC 検証に対応していることと，それらの 1/10 の，多くはアジア地区のクライアントが DNSSEC 検証に失敗する場合があるということであった．原因としては，DNSSEC での巨大な応答サイズの問題が考えられる．

Wang は 2013 年にクライアントクエリはジップの法則に従うという仮定を行ない，その積み重ねの結果が TLD DNS サーバに到達するという仮定を行い，推定した値と CN TLD のクエリ情報を比較し，仮定が正しいという結論を出した [56].

2.3.4 DNSへの攻撃及び攻撃検知に関する研究事例

Atkinsらは2004年にDNSの脆弱性の分析を行ない，RFC 3833 “Threat Analysis of the Domain Name System (DNS)” [8] としてまとめた．

Kaminskyは2008年に効率的なキャッシュポイズニング手法を示し，DNSプロトコルの問題点を指摘した[31]．提案手法の画期的な点は，攻撃のトリガーにランダムラベルを前置したクエリを用いることで，権威DNSサーバは不存在応答を返し，注入を試みるドメイン名そのものはキャッシュされずにクエリ名の不存在情報のみがキャッシュされることで，短時間での連続攻撃を可能としたことである．

Müllerは2008年に効果的なキャッシュポイズニング手法とその成功確率を示した[49]．

石原らは2014年にHadoopを利用したDNSトラヒックのセキュリティ解析をおこなう基盤について報告した[28]．彼らは，クエリ名を分析することで，ボットの検知などを試みている．

2.3.5 DNSプロトコルに関する改善

1983年に最初のバージョンが提案されたDNSプロトコル[44][45]は1987年に改定[46][47]され，IETFにて何度も拡張された．主な拡張は，1997年に標準化された動的な更新[55]とDNSプロトコルの明確化[22]，1998年に標準化されたネガティブキャッシュ[3]，1999年に標準化されたEDNS0拡張[53]，2005年に標準化されたDNSSECである．その後も多くの改善提案が行なわれており，一部が標準化された．

Wijngaardsは2008年にフルリゾルバ実装を改善し，キャッシュポイズニング攻撃への対策を行なう提案[60]を行ない，2009年に提案の改定[61]を行ない，Unboundに実装した．主にエントロピーの増大方法のまとめと，DNS応答中の応答セクション以外の追加情報をクライアントに応答

しないこと，CNAME などの応答に添付される情報を使用しないこと，委任情報を得た後に委任情報に対応する権威ある情報を取得するという提案である．

Vixie らは 2010 年にフルリゾルバ実装を改善し，委任情報の再検査などを行ない，キャッシュポイズニング攻撃への対策を行なう提案 [54] を行なった．

Contavalli らは 2011 年に公開フルリゾルバサービスによる DNS を使用した広域負荷分散への悪影響 [2] の対策として，フルリゾルバから権威 DNS サーバへのクエリにクライアントの IP アドレスの一部を添付する提案 [18] を行った．

Kumari らは 2014 年にフルリゾルバにルートのコピーを持たせ，ルート DNS サーバへのクエリを削減する提案を行った [36]．IETF での議論の結果，提案が改良され [35]，IETF DNSOP ワーキンググループで標準化対象とすることととなった．

Trenholme は権威 DNS サーバソフトウェア MaraDNS と，独自の名前解決アルゴリズムを採用した Deadwood というフルリゾルバを開発し，オープンソースソフトウェアとして公開している [52]．

2.4 トラフィック解析と DNS への影響

インターネットでは初期から各種統計情報を取得し，回線ごとに増強計画を立てるという活動が行なわれてきている．

Crowcroft らは 1991 年にイギリスの研究教育ネットワークからアメリカまでの国際線のパケットを調査し，プロトコルの使用割合や，トラフィックを多く扱う IP アドレスをはじめ，分析プログラムまで公開し，INET91 で報告した [19]．

Asaba らは 1992 年に日本の WIDE プロジェクトが運営するアメリカ

までの国際線のデータを収集し，プロトコルごとの回線使用度合などを INET92 で報告した [7].

長は 2010 年に IIJ が運用するブロードバンド接続サービスでのトラフィックを分析し，これまで安定して伸びてきたインターネットトラフィックが 2010 年 1 月に 20%近く急減したこと，その原因は著作権法改正による P2P 系アプリケーションのトラフィックの減少であること，その結果として HTTP のトラフィックが増えていることを報告した [33].

総務省は 2013 年に我が国のインターネットにおけるトラヒックの集計・試算を行ない，2010 年に 1 契約あたりのアップロード型トラフィックが減少し，そのあとも変化が少ないこと，ダウンロード型トラフィックは順調に増加し，5 年で 2 倍になっていることを示した [64].

長は 2014 年に IIJ が運用するブロードバンド接続サービスでのトラフィックを分析し，HTTPS による通信が着実に増加していることを示した [34].

HTTP や HTTPS のトラフィックが増加しているということは，大手コンテンツ事業者やコンテンツデリバリネットワークによるサービスが増加しているということであり，それらの事業者のほとんどが DNS を使用した広域負荷分散を行なっている．これは，基礎的なトラフィック計測が DNS 研究にとっても重要である事を示している．

2.5 DNS トラフィック解析の課題

DNS はインターネットにとり重要な機能であるため，多くの研究が行われてきた．権威 DNS サーバのクエリ解析，スタブリゾルバでのクエリ解析，フルリゾルバでのクエリ解析，DNS への攻撃及び攻撃検知，DNS プロトコルの改善に分類したが，その中には名前解決処理の問題点の指摘，新しいアプリケーションのための影響の調査などが含まれる．

先行研究により，ルートDNSサーバへのクエリが徐々に増加していることや，DNSクエリには本来不必要なものが多いこと，その原因はプログラムのバグによるものも多いということ，この状況は1992年から続いていることがわかる．このような状況にもかかわらず，インターネットは普及し，DNSの重要性は大きくなってきた．

ところが，クライアントからのクエリと権威DNSサーバへのクエリについて調査した研究が2002年のJungらによる研究[30]以後存在しない．その10年の間の変化を調査することは重要である．IPv6対応ノードは2001年には少なかったが，10年間のうちにIPv6対応OSが普及して，エンドノードのほとんどがIPv6に対応した．また，2012年6月6日にはWorld IPv6 Launchというイベントが行なわれ，大規模なサイト数百がIPv6サービスに対応した．さらに，コンテンツデリバリネットワークのような大規模な配信サービスが一般化し，DNSを用いた広域負荷分散を実装した結果，20や30，60，300といった小さなTTL値の使用が広まった．IPv6の影響について，Gaoらは2012年にAAAAクエリが増加していることを示したが，クライアントからのクエリ数については示していない[25]．DNSへの新しいアプリケーションでは，TTL値を0に近い値にすることが求められているものがあり，TTL値の現状を把握することも重要である．

そこで，第3章では，フルリゾルバ視点でみたIPv6クエリの変化と，応答タイプごとに分類したクライアントクエリごとに名前解決遅延，キャッシュヒット率，権威DNSサーバへのクエリ数を評価し，DNSへの負荷を減らす方法を提案する．さらに，存在しないTLDのクエリがルートDNSサーバへのクエリを多く生成していることを示す．第4章ではルートDNSサーバへのクエリについての詳細な解析とルートDNSサーバへの本来不必要なクエリを削減する方法を提案する．

第3章　フルリゾルバのキャッシュ効率と遅延改善

3.1　緒言

これまでDNSについて多くの研究が行なわれてきたが，新規サービスの展開により，DNSに関する新しい問題がみられるようになってきた．過去の研究と比べ，コンテンツデリバリネットワーク (Content Delivery Network, CDN) や大規模配信事業者によるDNSを使用した広域負荷分散の普及や，IPv6対応OSの普及による影響がでていると考えられる．

DNSを使用した広域負荷分散では，DNSクエリ送信元IPアドレスごとに応答するサーバのアドレスを変更したり，短時間で設定変更を行なうため，小さなTTL値を用いることが多い．TTL値を小さくするとキャッシュ可能な時間が短くなるため，DNSクエリが増え，クライアントへの遅延が大きくなる可能性がある．

さらに，IPv6普及活動として，2012年6月6日にWorld IPv6 Launchという活動が行なわれ，ISP，機器ベンダ，主要なウェブサービス事業者などが参加してIPv6でのサービスを有効化した．その結果，IPv6アドレスを問い合わせるタイプAAAAのクエリの増加と応答傾向の変化が見込まれる．

本章では，DNSについての主な懸念事項を3.2節に，データ収集方法と研究の契機となったデータの概要について3.3節に，懸念事項に関しての分析結果を3.4節に示し，問題点を認識し，3.5節にて改善点の指摘を

行なう．

3.2 DNSトラフィックに関する懸念事項

3.2.1 AAAA(IPv6アドレス) クエリの増加

10年前はIPv6対応のエンドノードはほとんど存在しなかった．しかしながら，現在のOSは標準でIPv6に対応しており，多くのユーザはIPv6のことを気にしないが多くのクライアント端末はIPv4アドレスを問い合わせるタイプAのクエリだけでなく，IPv6アドレスを問い合わせるタイプAAAAのクエリを送ることが推定され，AAAAクエリは10年間での純増となる．

ところが，2011年11月時点ではIPv6対応のサービスは非常に少なく，多くのサービスにはAリソースレコードだけ設定され，AAAAリソースレコードがなく，AAAAクエリの応答に値がないことが推定される．そのため，ドメイン名は存在するが値のない応答がDNSに及ぼす影響を評価する必要がある．

また，2012年6月6日に実施されたWorld IPv6 Launchは大規模なウェブサービスがIPv6サービスを有効化するイベントであったため，IPv4アドレスのみ設定されていた多くのサービスにIPv6アドレスが設定され，AAAAリソースレコードが設定された．その結果，AAAAクエリの多くでIPv6アドレスを含むAAAAリソースレコードの応答が得られるようになったと考えられる．応答が得られるとDNSへの影響が変わるため，World IPv6 Launch後の観測結果と変化を3.4.6節に示す．

3.2.2 大規模配信事業者でのDNSの使用

今日では，コンテンツデリバリネットワークなどの大規模配信事業者によるトラフィックが急速に増加している．大規模配信事業者はユーザから

のリクエストをユーザに近いサーバか，余裕のあるサーバに誘導しようと試みる．そのために，配信事業者は小さな TTL 値を用い，短時間にホスト名に対応する IP アドレスを変更する．また，配信事業者は顧客からみたサービスの簡素化のために，顧客には配信事業者のサービスホスト名への別名を示す CNAME リソースレコードを登録させ，顧客のサービス名から配信事業者のホスト名への別名展開をするようなサービスを提供しているところが多い．さらに，配信事業者や DNS ホスティング事業者では，ネームサーバホスト名を事業者のドメイン名のものとするサービス仕様の場合が多く，外部名のネームサーバホスト名を使用することが多い．サービス提供者の企業としてのドメイン名とサービス名が別の場合も，企業のドメイン名でサーバを提供する場合があり，外部名を使用している例がある．

小さな TTL 値の使用，CNAME の使用，外部名ネームサーバホスト名の使用は DNS の運用に対する 3 つの大きな懸念事項である．

3.2.3 小さな TTL 値

DNS 応答に含まれるリソースレコードは 32 ビットの正整数である TTL 値を持つ．TTL 値は，そのリソースレコードのキャッシュ可能時間を秒で示す．ゾーンの管理者は各 RRSet の TTL 値を指定する．ルートや TLD では，フルリゾルバのキャッシュが有効に働くように大きな TTL 値が設定されている．ルートや，com，net ゾーンのほとんどのデータの TTL 値は 2 日で，org や jp では 1 日である．

しかし，大規模配信事業者はドメイン名と IP アドレスの対応を頻繁に書き換えて広域負荷分散をおこなう．頻繁な変更のためには，フルリゾルバのキャッシュで保持される時間を短縮し，事業者の制御を短時間で反映するために TTL 値を小さくすることとなる．小さな TTL 値の使用は

フルリゾルバのキャッシュヒット率を下げると予想される．

しかし，事業者は各ゾーンの NS リソースレコードやネームサーバホスト情報の TTL 値，ネガティブキャッシュの TTL 値には比較的大きな値を使用し，最終的なアドレス情報を保持する DNS サーバまでの名前解決や，不存在情報のキャッシュヒット率は影響を受けない可能性があり，実態の把握には精密な計測を必要とする．

3.2.4 CNAME と外部名ネームサーバホスト名

CNAME リソースレコードは DNS の別名機能であり，もともとのサービス名から配信事業者のサーバにトラフィックを誘導するために使用される．

フルリゾルバが別名機能を示す CNAME リソースレコードに直面すると，別名先のドメイン名で名前解決をやり直すこととなる．もとのドメイン名と別名のドメイン名の TLD が異なっている場合は，ルート DNS サーバから名前解決をやり直す．CNAME の非効率的な使用は，ルート DNS サーバや TLD DNS サーバのクエリを倍増させる．また，フルリゾルバの資源，たとえばキャッシュをより多く使用する．

外部名ネームサーバホスト名の使用により，フルリゾルバや権威 DNS サーバへの負荷はさらに増加する．図 3.1 に，外部名ネームサーバホスト名を使用している場合の処理を詳細に示す．example.co.jp のネームサーバホスト名が外部名の ns1.example.com であるとし，エンドユーザが www.example.co.jp というドメイン名を入力したとする．まず，フルリゾルバはルート DNS サーバから名前解決を開始する (2)．次に，www.example.co.jp クエリを jp TLD DNS サーバに送る (4)．jp TLD DNS サーバは example.co.jp が外部名の ns1.example.com に委任されていることを知っているが， jp の外である ns1.example.com の情報を知らない．そのため，

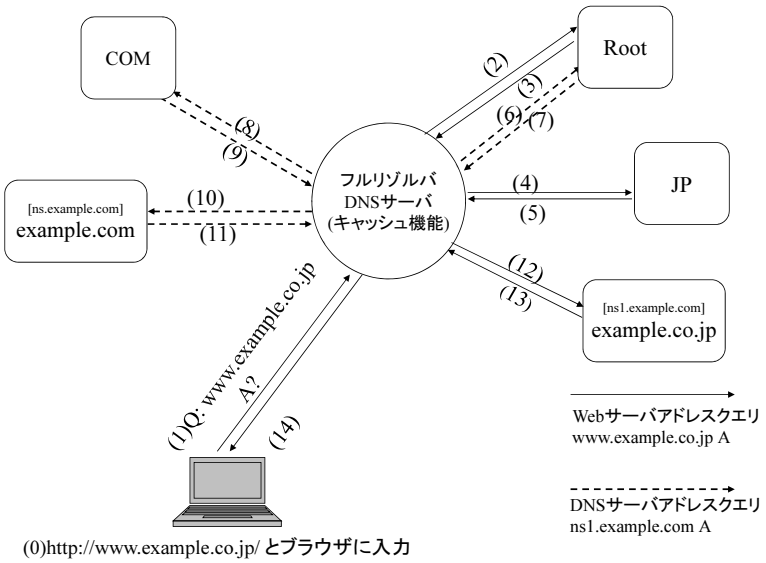

図 3.1: 外部名ネームサーバホスト名の例

フルリゾルバは ns1.example.com のアドレスの名前解決をルート DNS サーバから開始することになり，ルート DNS サーバ (6)，com TLD DNS サーバ (8)， example.com DNS サーバ (10) に順に ns1.example.com のタイプ A，AAAA のクエリを送る．最終的に，フルリゾルバは (11) で ns1.example.com の IP アドレスを受け取り，www.example.co.jp クエリを example.co.jp DNS サーバに送る (12).

結果として，外部名ネームサーバホスト名の使用はフルリゾルバの名前解決を複雑にし，フルリゾルバの資源の使用量を増やす．また，他の DNS サーバへの依存を増し，ルート DNS サーバ及び TLD DNS サーバの負荷を増す.

3.3　筑波大学でのデータ収集

AAAA クエリの増加と大規模配信事業者のクエリの影響は，ルート DNS サーバや TLD DNS サーバでは隠されることが多く，観測しにくい．この問題を調査するためには，ユーザ数が多いフルリゾルバか人気のあるウェブサイト，大規模なメールサーバなどを運用している組織の権威 DNS サーバでの観測をする必要がある．そこで，中規模のフルリゾルバとして筑波大学のフルリゾルバを選び，計測を実施した．

本節ではデータ収集環境と，取得データ中の異常値の扱い，取得したデータセットの概要を述べる．

3.3.1　データ収集環境

筆者らは筑波大学学情報環境機構学術情報メディアセンターが運用し，全学に提供するフルリゾルバのうち負荷の大きな 1 台の入出力パケットを収集し，分析することとした．筑波大学は中規模な ISP の規模があり，大学のフルリゾルバの分析により，DNS の最近の傾向と隠された問題を分析することができると考えられる．筆者らはエンドユーザのクエリと，フルリゾルバと権威 DNS サーバ間のパケットをすべて収集した．

データ収集環境を図 3.2 に示す．フルリゾルバの入出力の全パケットを，イーサネットスイッチでコピーし，データ収集 PC に送り，そこで収集した．収集されたデータは，(1) スタブリゾルバとフルリゾルバの間と (2) フルリゾルバから権威 DNS サーバのパケットを含む．

DNS クエリ情報は，エンドユーザのプライバシー情報を含むため，収集されたデータを 1 台のマシンにのみ蓄積し，限られた研究者にだけアクセス権限を与え，集計された統計情報だけをそのマシンから取り出すこととし，利用者のプライバシー情報を保護した．

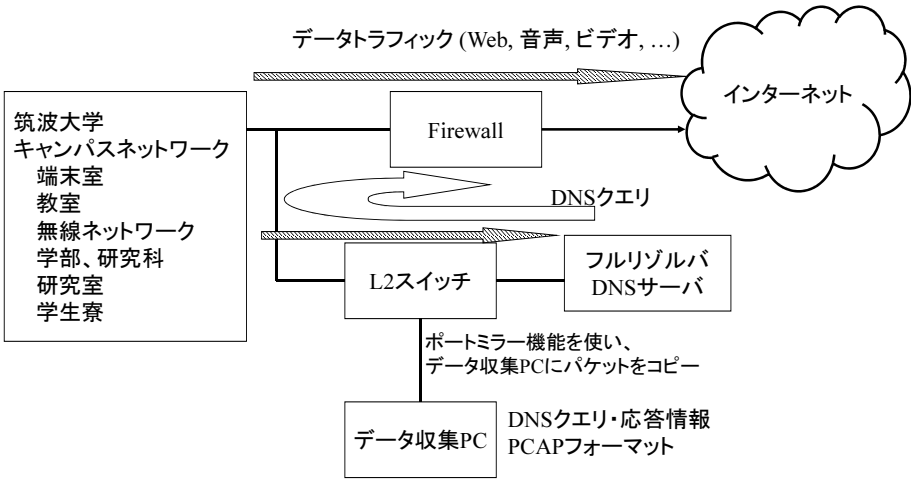

図 3.2: 筑波大学での DNS パケット収集環境

3.3.2 異常トラフィック

予備評価の結果，収集されたデータには異常なトラフィックが含まれることがわかった．図 3.3 はクエリ数が多いものから順に並べたクライアントからのクエリの積み上げグラフである．

最も多くのクエリを送るクライアントは，枚秒 5 回 localhost.localdomain クエリを送り，全体の 8%のクエリを送っていた．二番目のクライアントは，近い IP アドレス複数の逆引きクエリを毎秒送っていた．三番目と四番目はさまざまなクエリを送っていたため，正常と考えられる．そのため，一番と二番を一般性を欠く異常値とし，収集したデータから除外した．

3.3.3 収集したデータの概要

異常トラフィックを除外して，三つの 30 日づつのデータセットを作成した．表 3.1 にデータセットの概要と，比較のために Jung らによるデータセット [30] を示す．

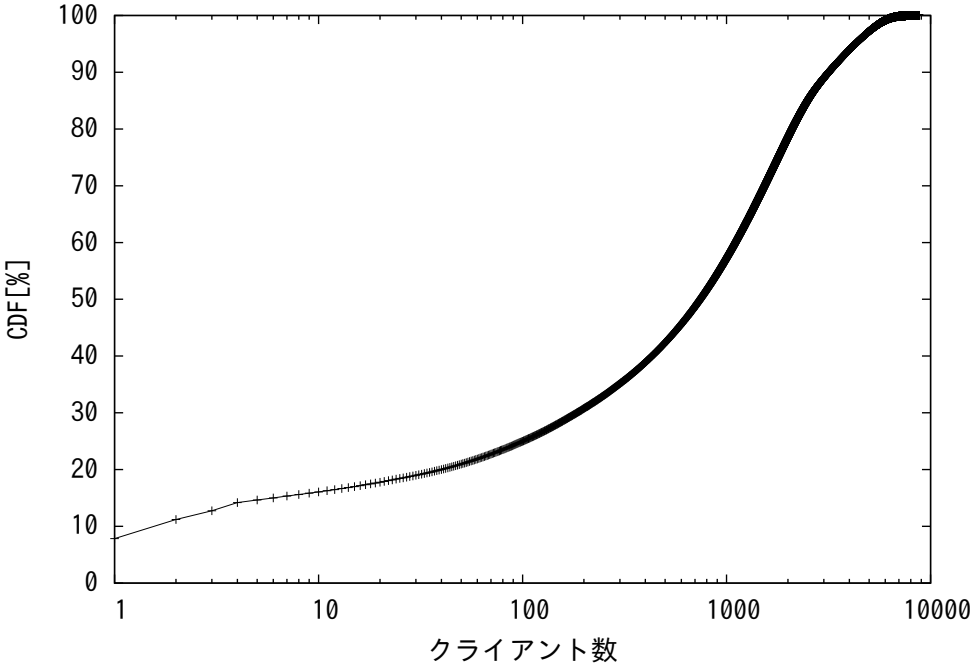

図 3.3: クエリ数順に並べたクライアントクエリの積み上げグラフ

表 3.1: DNS データセットの比較

Origin	[30]		Authors		
場所	MIT	KAIST	筑波大学		
年	2000	2001	2010	2011	2012
月/日	12/04	05/18	11/1	11/1	7/7
	-12/11	-5/24	-11/30	-11/30	-8/6
クライアントクエリ数	4,160,954	4,339,473	234,308,393	366,489,499	317,686,402
クエリ/秒	6.88	8.37	90.40	141.39	122.56
クエリ名数	302,032	219,144	3,375,088	4,015,966	2,971,084
クライアント数	1,216	8,605	6,556	8,815	11,662
クエリ/秒 /クライアント数			0.01378	0.01603	0.01051

データ収集期間や時代が違うため，直接の比較にはほとんど意味がないが，筑波大学の 2011 年 11 月のデータは，[30] のデータと比べ，クライアントからのクエリ数で 20 倍，クエリ名数で 10 倍以上で，クライアント数は同規模である．

我々のデータセットでは，2010 年 11 月と 2011 年 11 月を比べると，クライアント IP アドレス数が 56%増加し，クエリ数が 19%増加し，個々のクライアントからのクエリが毎秒 0.01378 から 0.01603 と 16%増加している．この増加は，DNS プリフェッチを実装したブラウザの普及を示していると考える．

2011 年 11 月と 2012 年 7 月を比較すると，クライアントクエリ数が 14%減少した．この減少は，大学のテスト期間と夏休みによるアクティビティの減少によるものと考える．

3.4 筑波大学のデータ分析

本節では，まず 2011 年 11 月のデータを分析し，次に 2012 年 7 月のデータセットと比較することで World IPv6 Launch の影響を示す．

3.4.1 指標の定義: キャッシュヒット率とクエリごとの TTL 値

キャッシュ機能は，クエリ応答時間を短縮するための最も重要な機能であるため，フルリゾルバのキャッシュヒット率を評価した．

スタブリゾルバであるクライアントからの一つの DNS クエリは，フルリゾルバから権威 DNS サーバへの複数のクエリを生成する．キャッシュヒット率を定義する前に，収集したパケット列から，エンドユーザのクエリから始まるすべての反復検索パケットと，最終的な応答までを含むクエリ列を抽出した．クエリ列は，CNAME による別名と，外部名ネー

ムサーバホスト名によるネームサーバホスト名クエリのクエリ列を含む．一つのスタブクエリは，ルート DNS サーバや TLD DNS サーバを含む複数の権威 DNS サーバへのクエリを生成する．

本章ではキャッシュヒット率を以下のように定義する．

$$\frac{\text{権威 DNS サーバへのクエリを生成しなかったスタブクエリ数}}{\text{スタブクエリ数}}$$

権威 DNS サーバへのクエリを生成したスタブクエリは，キャッシュヒットしなかったと分類する．

各々のスタブクエリは，複数の権威 DNS サーバへのクエリを生成し，複数の TTL 値を受信する．本章では，各スタブクエリごとに最小の TTL 値を求め，それを各クエリの最小 TTL 値とする．

3.4.2 キャッシュ効率

2011 年 11 月のデータセットを分析した結果を表 3.2 に示す．

第一列は，スタブクエリへの応答をクエリタイプ，CNAME 使用の有無，応答の状態，正引き，逆引き，TTL 値の違い，クエリ名の種類で分類したもので，詳細を表 3.3 に示す．第二列は，その行に分類されるデータの全クライアントクエリ中の割合である．第三列からのデータは，各行に分類されるデータでの指標の値である．第三列は，キャッシュヒット率である．第四列は，ルート DNS サーバへのクエリ数をクライアントからのクエリ数で除したもので，クライアントからの 1 つのクエリがルート DNS サーバに与えた影響を示す．第五列は，TLD DNS サーバへのクエリ数をクライアントからのクエリ数で除したもので，クライアントからの 1 つのクエリが TLD DNS サーバに与えた影響を示す．第六列は，ルート，TLD を含む権威 DNS サーバへのクエリ数をクライアントからのクエリ数で除したもので，クライアントからの 1 つのクエリが権威 DNS サー

表 3.2: キャッシュヒット率と権威 DNS サーバへの影響，遅延 (2011 年 11 月)

分類	全体の比率 [%]	キャッシュヒット率 [%]	権威 DNS サーバクエリ数 / クライアントクエリ数			フルリゾルバの名前解決時間 (平均) [ms]
			ルート	TLDs	全	
全体平均	100.0	75.1	0.00079	0.025	0.31	28.0
A	61.6	72.7	0.00100	0.037	0.36	30.7
AAAA	29.1	74.1	0.00012	0.007	0.27	28.8
CNAME あり	53.9	73.8	0.00070	0.025	0.34	30.4
CNAME なし	43.2	76.7	0.00096	0.028	0.26	24.9
存在応答	57.2	72.3	0.00071	0.039	0.37	31.4
Error	42.8	78.9	0.00090	0.007	0.22	23.4
·Server Failure	2.9	77.8	0.00007	0.002	0.54	89.5
·NoDATA	27.3	75.4	0.00006	0.007	0.24	23.4
· 名前不存在	11.5	93.5	0.00315	0.008	0.06	5.3
· 名前不存在 (TLD 不存在以外)	11.1	93.5	0.00008	0.008	0.06	5.3
·TLD 不存在	0.5	92.6	0.07303	0.000	0.07	5.4
正引き	92.4	73.5	0.00044	0.027	0.33	29.8
逆引き	7.3	95.7	0.00035	0.006	0.07	6.1
TTL 値 ≤ 300	44.0	67.3	0.00033	0.021	0.40	31.7
TTL 値 > 300	56.0	81.3	0.00115	0.029	0.23	25.1

バに与えた影響を示す．第七列は，クライアントからのクエリから応答までの，フルリゾルバの名前解決に要する時間の平均である．

全体平均のキャッシュヒット率は 75.1%であった．各々のクライアントクエリは，平均でルート DNS サーバへのクエリを 0.00079 回，TLD DNS サーバへのクエリを 0.025 回生成し，権威 DNS サーバへのクエリを 0.31 回生成する．権威 DNS サーバへのクエリ 0.31 回には，ルート DNS サーバへのクエリと TLD DNS サーバへのクエリを含む．また，クライアントとフルリゾルバの間の通信遅延を除いたクライアントからのクエリに対するフルリゾルバの名前解決に要する時間は平均 28ms である．

表 3.3: 応答の分類

分類	概要
全体平均	全データの平均
A	タイプ A(IPv4 アドレス) のクエリ
AAAA	タイプ AAAA(IPv6 アドレス) のクエリ
CNAME あり	応答に CNAME による別名を含む
CNAME なし	応答に CNAME を含まない
存在応答	スタブリゾルバが送信したクエリに対応する応答に，問い合わせたリソースレコードが含まれていた場合
Error	スタブリゾルバが目的のリソースレコードを得られなかったもの
Server Failure	フルリゾルバが応答コード 2 を返した場合で，設定エラーか，DNSSEC 検証のエラーにより，名前解決ができなかった場合
NoDATA	クエリ名は存在するが，タイプが一致するリソースレコードがない場合で，例えば，あるドメイン名が IPv4 アドレスを持っているが IPv6 アドレスアドレスを持たない場合にタイプ AAAA のクエリを送った場合が NoDATA となる．
名前不存在	クエリ名が存在しない場合
TLD 不存在	TLD が存在しないクエリ名の場合で，“名前不存在” は “TLD 不存在” を含む
逆引き	クエリ名の TLD が arpa であるもの
正引き	クエリ名の TLD が arpa 以外のもの
TTL 値 ≤ 300	応答の TTL 値が 300 以下であるもの
TTL 値 > 300	応答の TTL 値が 300 を越えるもの

名前解決中の CNAME の存在 (“CNAME あり”) はキャッシュヒット率を若干下げ，権威 DNS サーバへのクエリを少し増やす．しかし，違いは限られている．

小さな TTL 値 (≤ 300) は，キャッシュヒット率を下げるようにみえる．小さな TTL 値がコンテンツデリバリネットワークなどの影響を示すと考えられるため，3.4.5 にて詳細な評価を行なう．

その他の特徴を以下に示す．

- 57.2%のクライアントクエリの応答は “存在応答” であり，残り 42.8%の応答がなんらかの “エラー応答” である．“エラー応答” 時のキャッシュヒット率は 78.9%と “存在応答” の 72.3%よりも若干大きい．これは，“通常応答” の平均名前解決時間 31.4 ミリ秒と比較して短い，“エラー応答” 時の平均名前解決時間 23.4 ミリ秒に対応する．

- “Server Failure” は 89.5 ミリ秒と，非常に大きな平均名前解決時間であるが，全体の僅か 2.9%のクエリであるので，大きな問題にはなっていない．

- “名前不存在” と “TLD 不存在” のキャッシュヒット率は 93.5%と 92.6%と非常に高い．これらの高いキャッシュヒット率の結果，平均名前解決時間は非常に小さく 5.3 ミリ秒や 5.4 ミリ秒である．この結果は問題があることを示していない．しかし，ルート DNS サーバへのクエリ数に着目すると，“TLD 不存在” には問題があることがわかる．“全体平均” では，1 クライアントクエリは平均 0.00079 回のルート DNS サーバへのクエリを生成していたが，“TLD 不存在” の場合，全体平均の 92 倍の 0.07303 回のルート DNS サーバへのクエリを生成する．この事象を明確にするために，“名前不存在” から “TLD 不存在” を除外した “名前不存在 (TLD 不存在以外)” という項目によると，TLD が存在する “名前不存在” クエリは 1 クラ

イアントクエリあたり，0.00008 回のルート DNS サーバへのクエリとなり，全体平均の 1/10 と非常に小さな値となる．

- ルート DNS サーバへのクエリ数を調べたところ，そのフルリゾルバは一カ月で 289,492 のルート DNS サーバへのクエリを送っていた．そのうち，クライアントからのクエリの 0.5%である “TLD 不存在” がルート DNS サーバへのクエリの 45%を生成していた．

なお，フルリゾルバからの応答エラーの一つである “Server Failure” の場合，CNAME の存在は判断できないため，図 3.2 の CNAME ありの 53.9%，CNAME なしの 43.2%と “Server Failure” の 2.9%で 100%となる．A 列及び AAAA 列で 100%にならない理由は，それ以外のタイプのクエリが 9.3%存在するためである．正引き列と逆引き列で 100%にならない理由は，判断できないクエリ，例えば ”.” が存在するためである．

3.4.3 タイプ AAAA のクエリの増加

2001 年当時は，ほとんどの計算機の基本ソフト (Operating Systems, OS) は IPv6 に対応しておらず，IPv6 アドレスを問い合わせるタイプ AAAA のクエリはほとんどなかった．2011 年現在の OS，例えば Microsoft Windows Vista, Windows 7, Apple Mac OS X, Linux や BSD は IPv4 アドレスを問い合わせる A クエリと，IPv6 アドレスを問い合わせる AAAA クエリの両方をほぼ同時に送る．また，ほとんどの OS は IPv6 接続性がなくとも AAAA クエリを送る．2011 年時点で普及している Microsoft Windows は，IPv6 接続性がない場合でも，標準的に自動トンネル接続により IPv6 接続を作るような動作をし，そのために AAAA クエリを送ると考えられる．

図 3.4 に，2011 年 11 月に観測されたクライアントからのクエリタイプの比率を示す．62%が A(IPv4 アドレス) クエリであり，29%が AAAA(IPv6

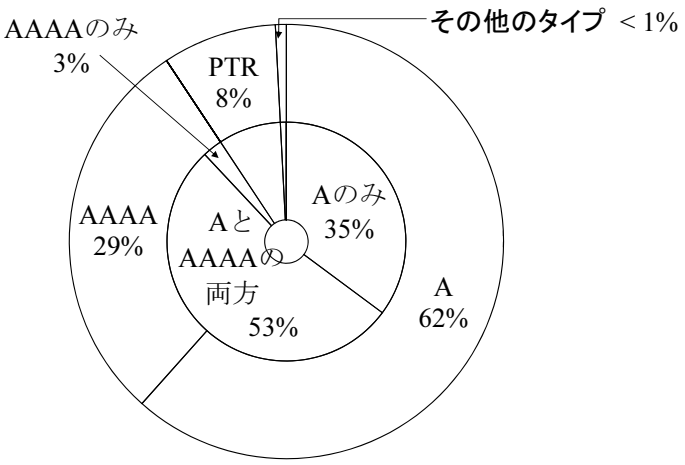

図 3.4: 2011 年 11 月のクライアントからのクエリタイプ

アドレス) クエリであった．53%のクエリが，A クエリと AAAA クエリを同時に発行したものであった．

そのため，10 年前と比較すると，IPv6 対応 OS の普及により，AAAA クエリが純増し，最大で 41% (29%/71%) クエリが増大したということができる．

ところで，8%を占めるタイプ PTR は，IP アドレスからドメイン名の対応を得るために使用されるタイプである．その他の 1%はそれ以外のタイプのクエリで，例えばメールサーバ情報を得る MX などのタイプである．

jp TLD オペレータである JPRS は，jp TLD の DNS サーバのうちの一台である A.DNS.JP のデータを収集しており，図 3.5 に，A.DNS.JP で観測したクエリ量の変化と，クエリタイプ A,AAAA の割合を示す．左側の縦軸は 2004 年 4 月からの jp クエリの増加率を示し，グラフには “+” で示す．jp TLD DNS サーバの増強が 2 度行なわれたため，増加率は折れ曲がっている．右側の縦軸はタイプ A のクエリとタイプ AAAA のクエリの割合を示し，2004 年に 7%であった AAAA クエリが 2011 年には 14%になっていることがわかる．しかし，フルリゾルバで観測された 29%と，jp

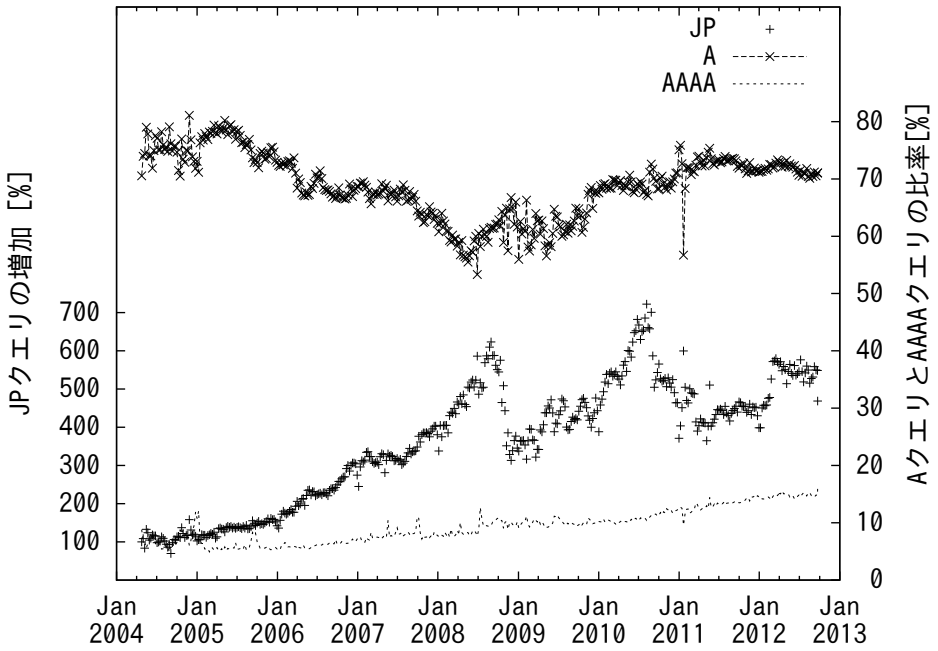

図 3.5: A.DNS.JP で観測した jp クエリの変化

表 3.4: ルートと jp TLD DNS サーバに送信する A クエリと AAAA クエリの数

送信先	クエリタイプ	一つのクライアントクエリが生成する権威 DNS サーバへのクエリ数
ルート	A	0.007559
ルート	AAAA	0.000116
ルート	すべて	0.004445
jp	A	0.043914
jp	AAAA	0.009056
jp	すべて	0.030236

TLD DNS サーバで観測された 14%は乖離している．

図 3.5 で A クエリの割合の変化が激しいが，A 及び AAAA 以外のクエリの割合が増えた時があると考えられ，具体的には jp TLD DNS サーバへの DoS 攻撃の影響でタイプ MX のクエリが急増した時期があったためと考えられる．そのため，A クエリの相対的減少は単調ではない．

AAAA クエリの増加を TLD DNS サーバで十分に観測できない理由を推定するために，フルリゾルバの入出力のクエリタイプを調査した．表 3.4 に，タイプ A, タイプ AAAA と，全タイプの 3 通りで，クライアントからのクエリがルート DNS サーバと TLD DNS サーバに及ぼす影響を示す．表 3.4 によると，調査対象フルリゾルバは A クエリと比較して，AAAA クエリのほうが，クライアントクエリによるルートや jp TLD DNS サーバへのクエリの生成が少ない．タイプ A ではルート DNS サーバに 0.005559 回，jp TLD DNS サーバに 0.043914 回であるが，AAAA クエリではルート DNS サーバに 0.000116 回，jp TLD DNS サーバに 0.009056 回と非常に少ない．

表 3.5: A クエリと AAAA クエリの送り方の違いによるクライアント数

	2011 年 11 月		2012 年 7 月	
場合	クライアント数	%	クライアント数	%
全クライアント数	8815	100.0	11662	100.0
A クエリを送ったもの	8707	98.8	11588	99.4
AAAA クエリを送ったもの	7772	88.2	10247	87.9
A と AAAA クエリを同時に送ったもの	7720	87.6	10202	87.5
同時に送ったもののうち				
A の後に AAAA を送ったもの	6993	79.3	9303	79.8
AAAA の後に A を送ったもの	100	1.1	160	1.4
順序が混ざっていたもの	627	7.1	739	6.3

表 3.5 に，クライアントからの A クエリと AAAA クエリの送り方の違いを示す．

表 3.5 の 2011 年 11 月のデータによると，88.2%のクライアントが A クエリと AAAA クエリの両方を送るが，79.3%が A クエリを先に送る．A クエリと AAAA クエリの間の時間が，フルリゾルバでの A クエリの名前解決時間よりも長ければ，ルート DNS サーバと TLD DNS サーバからの委任情報は AAAA クエリが到達する前にキャッシュされる．なお，A クエリをルート DNS サーバや TLD DNS サーバに送った時に得られる委任情報は，AAAA クエリを送った時に得られる委任情報と同じである．委任情報がキャッシュされている場合，フルリゾルバはルート DNS サーバや TLD DNS サーバに AAAA クエリを送らず，事前の A クエリによりキャッシュされた委任情報を用いて，AAAA クエリを各組織の権威 DNS サーバに直接送る．79.3%のほとんどのクライアントが A クエリを先に送るため，キャッシュ機構がルート DNS サーバや TLD DNS サーバで観測される AAAA クエリを削減している．つまり，クライアントで観測された 29%の AAAA クエリの増加を jp TLD DNS サーバでは 14%しか観

測できない理由は，多くのクライアントがA クエリを先に送るからであると考えられる．

クライアントがA クエリを先に送り，若干の遅延のあとAAAA クエリを送る場合のクライアントとフルリゾルバの動作を図3.6 に示す．その場合，ユーザがブラウザに “http://www.example.jp/” と入力すると，クライアントはフルリゾルバにwww.example.jp A クエリを最初に送り，次にwww.example.jp AAAA クエリを送る．クライアントからフルリゾルバへのAAAA クエリの送信がフルリゾルバによるA クエリの名前解決の進展である図3.6 の (1-1) から (1-5) よりも後であれば，クライアントからAAAA クエリが届くより前にjp と example.jp の情報がキャッシュされる．その後，フルリゾルバはクライアントから www.example.jp AAAA クエリを受け取るが，以前のタイプA のクエリでキャッシュされたjp とexample.jp の情報を持っているため，ルートDNS サーバとTLD DNS サーバには www.example.jp AAAA クエリを送らず，www.example.jp AAAA クエリを直接example.jp のサーバに送る．結果として， www.example.jp AAAA クエリは，フルリゾルバと example.jp 権威DNS サーバでは観測されるが，ルートDNS サーバとjp TLD DNS サーバでは観測されない．

AAAA クエリを先に送るクライアントや，A クエリとAAAA クエリの送出時間差が小さい場合には，ルートDNS サーバやTLD DNS サーバでもAAAA クエリが観測される．

なお，表3.5 の2012 年7 月のデータセットについては3.4.6 節で紹介する．

A，AAAA，PTR，その他のクエリタイプについて応答タイプを分類したものを表3.6 に示す．

“名前不存在” は，クエリ名が存在しない場合である．“NoDATA” は，クエリ名は存在するが，クエリタイプが存在しない場合である．“Server Failure” は，設定エラーなどによるエラーかDNSSEC 検証エラーである．

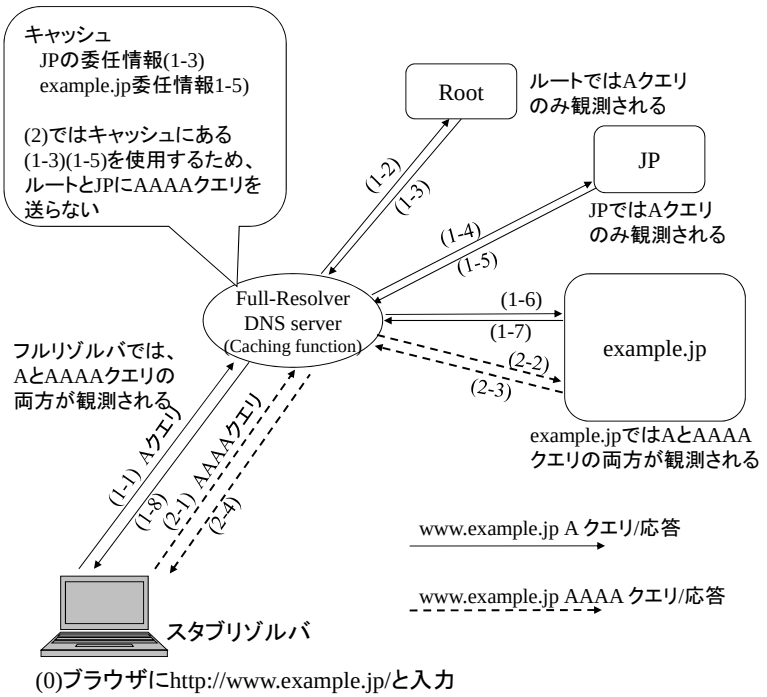

図 3.6: A クエリと AAAA クエリ両方を送るクライアント

表 3.6: 2011 年 11 月の応答タイプ

クエリタイプ	合計	A	AAAA	PTR	その他
クエリ数	366.5	225.7	106.6	31.1	3.0
[100 万]	100%	62%	29%	8%	1%
応答の種類	%	%	%	%	%
Server Failure	2.9	0.6	1.8	23.3	1.3
名前不存在	11.5	8.1	3.4	59.1	67.7
Refused	0	0	0	0	0.3
存在応答	57.2	89.3	2.0	17.4	17.1
NoDATA	27.5	0.7	92.6	0.1	12.2
Timeout	0.9	1.3	0.2	0.1	1.4

“Timeout” は，フルリゾルバが応答を返さなかったものである．“Refused” は，フルリゾルバが RCODE 1 を返す場合で，応答しないという設定による．“名前不存在” と “NoDATA” は，ネガティブキャッシュTTL 値に指定された時間，キャッシュすることができる．“Server Failure”，“Time out”，“Refused” はキャッシュすることができない．

表 3.6 によると，A クエリの 89.3%が “存在応答” であり，AAAA クエリの 92.6%は “NoDATA” 応答で，A と AAAA で明確に異なる．これは IPv6 アドレスを使用するサービスが少ないためで，わずか 2.0%の AAAA クエリが IPv6 アドレスを受信していることを示す．

AAAA クエリについて主な発見のまとめは以下の通りである．

1. クライアントの 88.2%は IPv6 対応しており，同じクエリ名の A クエリと AAAA クエリを連続してフルリゾルバに送る．また，79.3%のクライアントは，A クエリを先に送り，そのあとに AAAA クエリを送る．A クエリを先に送るという順序とフルリゾルバのキャッシュの効果により，ルート DNS サーバや TLD DNS サーバでは AAAA クエリの増加を部分的にしか観測できない．AAAA クエリの増加はフルリゾルバで明確に観測できる．

2. IPv6 に対応したサービスは少なく，ほとんどの AAAA クエリは “NoDATA” 応答となる．多くの “NoDATA” 応答では，ネガティブキャッシュの TTL 値が対応する A リソースレコードの TTL 値と同じか大きい．“NoDATA” 応答はキャッシュされ，権威 DNS サーバへの影響は “存在応答” と同じである．

IPv6 対応サービスは増加しつつあるため，“NoDATA” 応答は徐々に “存在応答” に変わっていくと考えられる．しかし，キャッシュ機構と現在のクライアントのクエリ順序により，ルート DNS サーバや TLD DNS サーバでは AAAA クエリの増加を観測しにくく，応答の変化は観測できない．

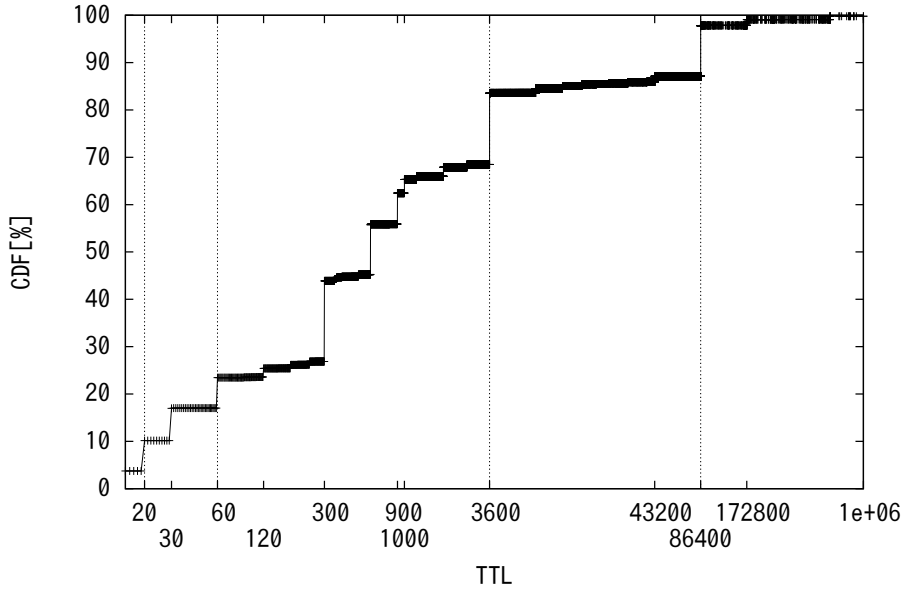

図 3.7: 応答の最小 TTL 値の分布

3.4.4 小さな TTL 値の影響

表 3.2 で示した通り，TTL 値はキャッシュヒット率に影響を与える．追加の評価として，TTL 値の累積分布グラフを図 3.7 に示す．

図 3.7 によると，非常に小さな TTL 値がよく使われており，6.4%が 20 秒，6.8%が 30 秒である．TTL 値が 31 未満の応答を受け取るクライアントクエリの量は全体の 14.0%であり，これらはかなりの負荷を DNS に与える．これらの小さい TTL 値は，大規模なコンテンツ提供者や大規模配信事業者に利用されている．例えば，2012 年 7 月には，Akamai は TTL 値として 20 を使用し，twitter.com と www.facebook.com は 30，Yahoo は 60，Google は 300 を使用していた．

次に，クエリ頻度と TTL 値ごとのキャッシュヒット率を図 3.8 に示す．

クエリ頻度は，クエリ名・クエリタイプの組ごとに一カ月のクエリ数

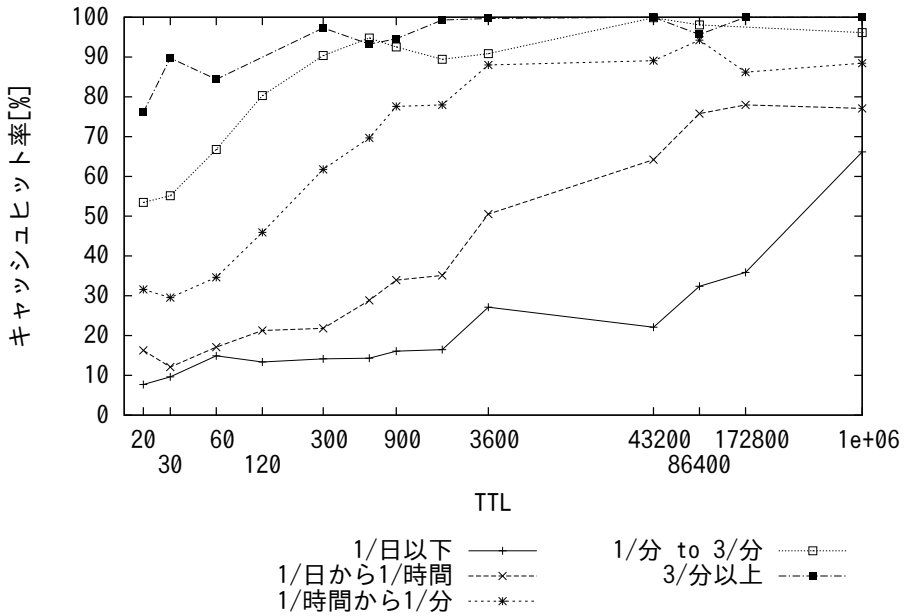

図 3.8: 平均クエリ頻度と TTL 値ごとのキャッシュヒット率

を平均し，5 つに分類した．それぞれ TTL 値 86400，3600，60，20 を境に影響が変化すると考えられ，図 3.7 に縦線を追加した．

- 1 日に 1 クエリ以下
- 1 日に 1 クエリを超え，1 時間に 1 クエリ以下
- 1 時間に 1 クエリを超え，1 分に 1 クエリ以下
- 1 分に 1 クエリを超え，1 分に 3 クエリ以下
- 1 分に 3 クエリを超えるもの

小さな TTL 値のクエリはキャッシュヒット率が小さくなると期待したが，図 3.8 によると高頻度のクエリがある場合は TTL 値が小さくてもキャッシュヒット率が大きい．例えば，TTL 値 20 で，頻度が一分間に 3 クエ

リを超える場合はキャッシュヒット率 76.1%である．大規模配信事業者は頻繁なアクセスがあるドメイン名を保持しているため，小さな TTL 値によるキャッシュヒット率の低下が隠蔽される．しかし，高頻度のクエリによるキャッシュヒット率の増大は DNS の問題を単純化するわけではないことに注意が必要であり，不要な高頻度クエリは存在そのものが問題である．

TTL 値を 20 から 300(5 分) に変更することで，例えば一分に 3 回を超えるクエリ頻度の場合で 28%(76.1%から 97.3%)，キャッシュヒット率を増加させることができる．適切に大きな TTL 値の設定がフルリゾルバと権威 DNS サーバの負荷の軽減につながるため，配信事業者などのインターネットサービスプロバイダはサービスに適した TTL 値を選ぶことが必要である．配信事業者が滅多にアクセスがないドメイン名に小さな TTL 値を設定すると，顧客は低いキャッシュヒット率による名前解決時間の増加，つまり，ドメイン名を入力してから表示されるまでの遅延の増大に直面する．

なお，表 3.2 に示した通り，“TLD 不存在” なクエリは多くのクエリをルート DNS サーバに送り，ルート DNS サーバへ送られるクエリの 45%を占めたため，分析に影響を与える．そのため，図 3.8 と図 3.9(3.4.5 節) では “TLD 不存在” となるクエリと “名前不存在” となるクエリを除外した．

3.4.5 CNAME リソースレコードの効果と，外部名ネームサーバホスト名の影響

本節では，3.2.2 節で述べた CNAME の使用と外部名ネームサーバホスト名の使用による問題について詳細に述べる．

図 3.9 に，CNAME による別名変換の有無によるクエリ頻度ごとのキャッシュヒット率と，権威 DNS サーバへのクエリ数を示す．CNAME の使用はキャッシュヒット率を約 10%下げ，ルート DNS サーバと TLD DNS サー

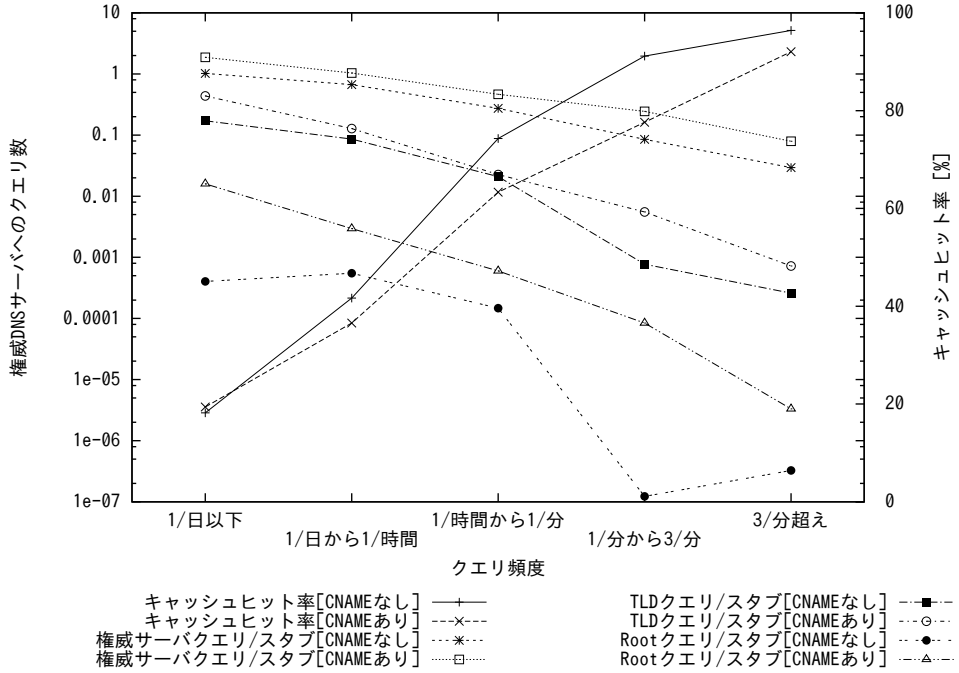

図 3.9: CNAME 使用による，クエリ頻度ごとのキャッシュヒット率と権威 DNS サーバへのクエリ数

バへのクエリ数を増加させる．また CNAME の使用は権威 DNS サーバへのクエリ数を倍増させる．図 3.9 での最も頻繁にアクセスされる頻度である一分に 3 回を超える場合でも，CNAME の使用はルート DNS サーバへのクエリを 10 倍に増やし，TLD DNS サーバへのクエリを 2.8 倍に増やす．

図 3.8 で示した平均クエリ頻度と TTL 値ごとのキャッシュヒット率と同様に，高頻度のクエリはキャッシュヒット率を増大する．例えば，CNAME を使用している場合でも，1 分あたり 1 ～ 3 クエリのクエリ頻度であれば 75%のキャッシュヒット率である．しかし，同じ頻度で CNAME を使用しないクエリ名の場合はキャッシュヒット率 92%となり，75%よりも非常によい値となる．差分である 17%は，CNAME 使用によるパフォーマンス

表 3.7: 外部名ネームサーバホスト名使用によるキャッシュヒット率と権威 DNS サーバへの影響

分類	全体の比率 [%]	キャッシュヒット率 [%]	権威 DNS サーバクエリ数 / クライアントクエリ数			フルリゾルバの名前解決時間 (平均) [ms]
			ルート	TLDs	全	
全体平均	100.0	75.1	0.00079	0.025	0.31	28.0
すべてのネームサーバホスト名が内部名	27.5	78.1	0.00035	0.009	0.24	22.5
すべてのネームサーバホスト名が外部名の委任あり (.arpa 以外)	60.1	71.5	0.00042	0.032	0.36	33.1
一部のネームサーバホスト名が外部名の委任あり	3.7	69.4	0.00118	0.039	0.40	28.6
逆引き (.arpa)	7.3	95.7	0.00035	0.006	0.07	6.1
分類不能	1.4	81.2	0.02559	0.116	0.29	31.5

の低下である．このようなパフォーマンスの低下は，一日に 1 クエリ以上のアクセス頻度の場合にみられる．

外部名ネームサーバホスト名と内部名ネームサーバホスト名の使用によるキャッシュヒット率の違いを表 3.7 に示す．IP アドレスからホスト名への逆引き DNS では，地域インターネットレジストリ (Regional Internet Registries, RIRs) の登録規則や登録システムが内部名ネームサーバホスト名の登録を想定しておらず，必ず外部名ネームサーバホスト名を使用することから，CDN などの配信事業者の影響と分離するために，“ip6.arpa” ドメイン名と “in-addr.arpa” ドメイン名を使用する逆引き DNS を “逆引き” として独立して分類した．

表 3.7 は，今日のインターネットでは外部名ネームサーバホスト名が一般的に使用されており，全クエリの 67.4%(60.1%+7.3%) が，すべてのネー

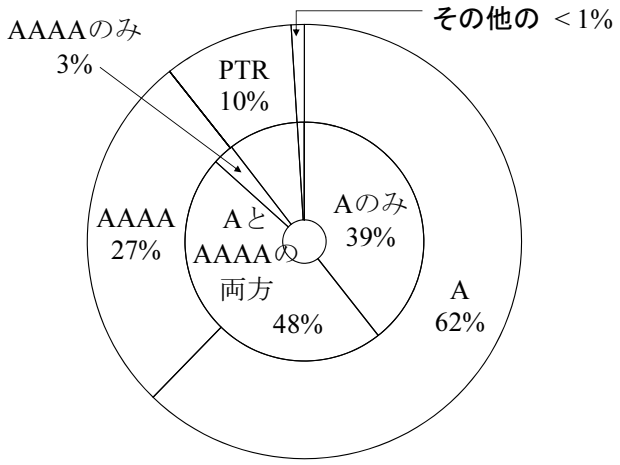

図 3.10: 2012 年 7 月のクライアントからのクエリタイプ

ムサーバホスト名が外部名である委任点のあるクエリ名であることを示す．すべてのネームサーバホスト名が外部名である委任点を持つクエリ名の名前解決は，すべての委任点のネームサーバホスト名が内部名であるクエリ名の名前解決に比べ，ルート DNS サーバへのクエリを 20%(0.00035 から 0.00042) 増加させ，TLD DNS サーバへのクエリを 3.7 倍 (0.009 から 0.032) 増加させる．

内部名ネームサーバホスト名と比べた外部名ネームサーバホスト名の最も重要な性能低下は，名前解決処理時間の 50%増加 (22.5 ミリ秒から 33.1 ミリ秒) である．名前解決処理時間の増大は，ユーザからみた待ち時間の増大であるため，影響が大きい．

3.4.6 World IPv6 Launch による変化

図 3.10 に 2012 年 7 月のクライアントからのクエリタイプの割合を示す．

図 3.10 と図 3.4 を比較すると，2011 年 11 月と 2012 年 7 月で，クライアントの動作に変化がないことがわかる．

表 3.8: 2012 年 7 月のデータセットでの応答タイプ

クエリタイプ	合計	A	AAAA	PTR	その他
クエリ数 $[*10^6]$	317.7	200.2	84.3	30.1	3.1
	100%	63%	26.5%	9.5%	1%
応答タイプ	%	%	%	%	%
Server Failure	2.1	0.9	1.2	12.4	1.0
名前不存在	17.3	14.8	5.1	64.7	47.3
Refused	0	0	0	0	0.0
存在応答	60.7	83.4	21.2	22.8	31.6
NoDATA	19.6	0.6	71.9	0.1	4.3
Timeout	0.4	0.3	0.1	0.1	15.6

表 3.5 と図 3.5 も同様に，World IPv6 Launch の前後の 2011 年 11 月と 2012 年 7 月とでクライアントの動作に変化がないことを示す．

表 3.8 に，2012 年 7 月の応答タイプを示す．

表 3.8 と表 3.6 を比較すると，“名前不存在” が 11.5%から 17.3%に増加し，“NoDATA” が 27.5%から 19.6%に減少していることがわかる．顕著な変化は AAAA クエリの応答に見られ，IPv6 アドレスを返す “AAAA 存在応答” が 2.0%から 21.2%に増加し，IPv6 アドレスを持たない “AAAA NoDATA” は 92.6%から 71.9%に減少した．World IPv6 Launch という活動で，数百の有名な WWW サイトが AAAA リソースレコードの設定を行ない，AAAA クエリの 21.2%に IPv6 アドレスが応答したと考えられる．

表 3.9 は，2012 年 7 月のデータセットでのキャッシュヒット率，権威 DNS サーバへのクエリ数，平均名前解決時間を示す．

表 3.9 を 2011 年 11 月のデータである表 3.2 と比較すると，差分は僅かでほとんど同じ傾向を示し，以下の違いがあった．

表 3.9: キャッシュヒット率と権威 DNS サーバへの影響，遅延 (2012 年 7 月)

分類	全体の比率 [%]	キャッシュヒット率 [%]	権威 DNS サーバクエリ数 / クライアントクエリ数 ルート	TLDs	全	フルリゾルバの名前解決時間 (平均) [ms]
全体平均	100.0	75.6	0.00063	0.020	0.29	26.8
A	63.0	74.1	0.00071	0.029	0.34	29.0
AAAA	26.5	71.6	0.00017	0.005	0.28	29.9
CNAME あり	55.2	71.9	0.00052	0.020	0.36	30.8
CNAME なし	44.4	80.9	0.00077	0.020	0.21	20.6
通常応答	60.7	73.0	0.00049	0.030	0.35	29.5
Error	39.3	79.7	0.00084	0.005	0.21	22.6
·Server Failure	2.1	59.4	0.00009	0.002	0.83	109.0
·NoDATA	19.5	69.2	0.00006	0.006	0.29	30.1
· 名前不存在	17.3	96.2	0.00183	0.004	0.04	3.0
· 名前不存在 (wo TLD 不存在)	16.8	96.3	0.00004	0.004	0.04	2.9
·TLD 不存在	0.5	93.5	0.06387	0.000	0.06	5.6
正引き	91.7	73.6	0.00033	0.021	0.32	29.0
逆引き	7.9	97.9	0.00019	0.002	0.03	2.3
TTL 値 ≤ 300	46.1	65.8	0.00024	0.017	0.41	32.4
TTL 値 > 300	53.9	84.1	0.00097	0.022	0.19	22.0

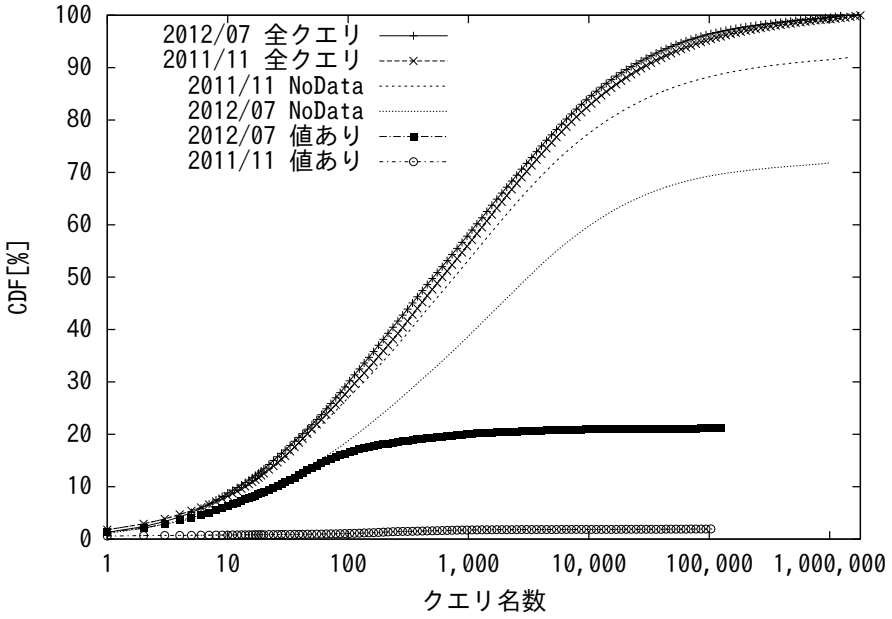

図 3.11: World IPv6 Launch 前後の AAAA クエリのクエリ名の累積比率分布

- 全体での権威 DNS サーバへのクエリ数が 0.31 から 0.29 と若干減少した
- AAAA クエリのキャッシュヒット率が 74.1%から 71.6%と若干減少した
- AAAA クエリによる，権威 DNS サーバへのクエリ数が 0.28 から 0.29 と若干増加した

World IPv6 Launch 前後の 2011 年 11 月と 2012 年 7 月のデータセットを用い，IPv6 アドレスを問い合わせる AAAA クエリについて，クエリ名ごとにクエリ数が多い順に並べた累積比率分布を図 3.11 に示す．

“全クエリ” と，IPv6 アドレスが存在する “値あり”，“NoDATA” となるものをプロットした．“全クエリ” については，2011 年 11 月のデータ

と 2012 年 7 月のデータはほぼ同じ傾向を示し，クエリ名数そのものもほぼ同じ数である．World IPv6 Launch 前には IPv6 アドレスを持つ “値あり” のクエリ名数は 10 万以上であったが，クエリの割合は 2.0%であった．World IPv6 Launch 後には，IPv6 アドレスを持つ “値あり” の上位 100 のクエリ名が 16.5%のクエリを占め，上位 1000 のクエリ名が 20.0%のクエリを占めている．

IPv6 アドレスを持つ AAAA リソースレコードを返したホスト名で最もクエリが多かったものは，www.facebook.com ，dns.msftncsi.com と Google のサービスで使用されるドメイン名であった．dns.msftncsi.com は， Microsoft Windows でネットワークのテストのために使用されるドメイン名である．

World IPv6 Launch の影響をまとめると，World IPv6 Launch 後に数百の人気のあるドメイン名が IPv6 アドレスを設定し， AAAA リソースレコードを返すようになったことで，AAAA クエリの 21.2%が IPv6 アドレスを返すようになったことである．しかし，DNS の動作への影響はなかった．

3.5 非効率的な DNS 設定への対策

図 3.4 で示した通り，今日のクライアント端末は IPv6 接続性がない場合でも IPv6 アドレスクエリ (タイプ AAAA) を送る．この動作はすでにフルリゾルバの負荷を上げ，権威 DNS サーバの負荷を上げている．インターネット利用者はこの標準の動作を変更し，AAAA クエリを禁止することができるが，ほとんどの利用者は標準設定を変更しない．そこで，もし，基本ソフト (OS) 提供者が標準設定を変更し，IPv6 接続性がない場合には AAAA クエリを送らないこととすると，筑波大学の場合はフルリゾルバのクエリを 29%削減することができる．さらに，アクセスライン

のトラフィックも減らすことができる．

適切に大きなTTL値の選定は，クライアントからみた名前解決遅延を短縮する．

3.4.5節で述べた通り，外部名ネームサーバホスト名の使用は，クエリ応答時間の顕著な増大を招く．また，ルートDNSサーバとTLD DNSサーバへのクエリ数を増大させる．この不適切な設定は，WWWサーバ管理者などのDNSに関する不十分な理解が原因であると考えれられる．しかし，外部名ネームサーバホスト名は，同じIPアドレスに対して新しい内部名の名前を作成し，外部名から内部名に変更することで容易に修正できる．この手法はよく知られている．図3.1を例に外部名ネームサーバホスト名を内部名に変更する効果を推定すると，点線のネームサーバホスト名クエリが削減されるため，権威DNSサーバへのクエリを50%削減できることとなる．また，表3.7によると，この対策でDNSの応答遅延を33.1ミリ秒から22.5ミリ秒に改善できる．

不要なCNAMEの削除は，クライアントからみた名前解決の応答時間を平均で30.4ミリ秒から24.9ミリ秒に減らすことができる．CNAMEによる別名は，サービスの設計やわかりやすさという点で重要ではあるが，性能劣化を考慮する必要がある．CNAMEリソースレコードの所有者名とその別名のドメイン名が同じゾーン内に含まれる場合，そのCNAMEリソースレコードはルートDNSサーバ及びTLD DNSサーバへのクエリ数を増加させない．たとえば，“www.example.jp. IN CNAME www.l.example.jp.”というCNAMEの所有者名と別名が “example.jp” のサブドメイン名の場合である．この例のように，別名をもとの所有者名と同じか同じドメイン名のサブドメイン名とした場合には，ルートDNSサーバとTLD DNSサーバからの応答はキャッシュされたものを使用することができ，ルートDNSサーバとTLD DNSサーバへのクエリを削減することができる．

サーバ管理者は，外部名ネームサーバホスト名の使用，CNAMEの使

用，小さなTTL値の使用という問題を修正することができ，修正することでルートDNSサーバとTLD DNSサーバへのクエリを削減するだけでなく，エンドユーザの名前解決遅延時間を短縮できる．

3.6 結言

本章では，今日のIPv6の普及とコンテンツデリバリネットワークなどの大規模配信事業の普及による影響を調べるため，筑波大学のフルリゾルバのクエリ分析を行ない，キャッシュ効率増大や遅延低減のための適切な設定の提案を行なった．主な結論を以下にまとめる．

(1) 筑波大学のフルリゾルバのクエリ分析

本章では，筑波大学情報環境機構学術情報メディアセンターの運用するフルリゾルバの入出力パケットを分析した．そのフルリゾルバは全学のユーザを対象としており，学生寮やWi-Fi 接続の通常のPCが参照するため，利用帯域等のデータから中規模ISPのフルリゾルバに近いと考えられる．時期によるが一ヶ月で8800から11000のクライアントIPアドレスを観測した．

キャプチャデータの分析の結果，IPv6対応OSの普及によりIPv6アドレス(AAAA) クエリが増加したため，クライアントからのクエリが41%増加したこと，CDNなどのDNSを使用した広域負荷分散サービスの普及により，小さなTTL値，CNAME，外部名ネームサーバホスト名の使用が増えていることがわかった．結果として，エラー応答の割合の増加やキャッシュヒット率の低下，応答遅延の増大がみられた．

クライアントからのAAAAクエリによる権威DNSサーバへのクエリはよくキャッシュされ，存在応答と同じキャッシュ効率であり，ク

ライアントからのAAAAクエリによるルート，TLD DNSサーバへのクエリの増加は観測されなかった．

また，2012年6月6日のWorld IPv6 Launchの前後でAAAAクエリに存在応答が戻る割合が2.0%から21.2%に増加した．これは，World IPv6 Launch以降，よく使用されるドメイン名のIPv6対応が進んだことを示している．

(2) キャッシュ効率増大や遅延低減のための適切な設定の提案

(1) の結果は，(a) 無用なAAAAクエリの低減，(b) 適切に大きなTTL値の設定，(c)CNAMEの削減，(d) 内部名ネームサーバホスト名の適切な設定，が重要な運用課題である事を示している．これらの結果を背景に，本章では以下の指摘と提案を行った．

(a)IPv6対応OS(比較的新しいバージョンのMicrosoft WindowsやApple Mac OS X，Linuxなど) はIPv6接続性がない場合でもAAAAクエリをフルリゾルバに送信する．これらIPv6対応OSが，IPv6接続性の不存在を検知し，不要なAAAAクエリを省略することで，フルリゾルバの負荷を軽減することができる．

(b)DNSを利用した広域負荷分散で使用される小さなTTL値は，キャッシュヒット率を下げ，フルリゾルバの負荷を上げ，権威DNSサーバへのクエリ数を増大させる．名前解決にかかる平均時間も，TTL値300以下で31.7ミリ秒が，TTL値が300を超える場合には25.1ミリ秒と，TTL値が大きい方が短い．適切に大きなTTL値を使用する事で名前解決に要する時間を短縮できる．

(c)CNAMEを使用する複雑なドメイン名の名前解決に要する平均時間は30.4ミリ秒であったが，CNAMEを使用しないドメイ

ン名の場合は 24.9 ミリ秒であった．CNAME の使用を最小限にすることで名前解決に要する時間を短縮できる．

(d) 外部名ネームサーバホスト名を使用するドメイン名の名前解決に要する時間は平均 33.1 ミリ秒であったが，すべて内部名ネームサーバホスト名を使用するドメイン名の場合は平均 22.5 ミリ秒であり，内部名ネームサーバホスト名を適切に使用することで名前解決の時間を短縮できる．

これらの施策を行なうことで，フルリゾルバの負荷を低減でき，名前解決に要する時間を短縮できる．

DNSSEC [4] は最近追加された DNS への重要な変更である．DNSSEC の普及はまだ進行中であるため，本章では DNSSEC 普及に伴う問題については分析しなかった．近い将来，DNSSEC 普及による影響が重要な問題になる．

第4章　ルートDNSサーバへのクエリ数の削減

4.1　緒言

ルートDNSサーバは，フルリゾルバでの名前解決の根幹であり，全世界のフルリゾルバからのクエリを受ける．そのため，ルートDNSサーバには莫大なクエリが集中しており，ルートDNSサーバの設備増強などを計画するためにはルートDNSサーバ側でのクエリ解析が重要である．また，ルートDNSサーバには不必要なクエリが多いという先行研究 [17][43][15][16] [38] があり，その実態を評価するためにはルートDNSサーバでのクエリ解析だけでなくフルリゾルバ側での原因の追究が必要であり，原因解明後，フルリゾルバの名前解決アルゴリズムを直すことで，ルートDNSサーバへの負荷を減らすことができる．

本章では，第4.2節で，ルートDNSサーバのクエリ情報をもとにIPアドレスごとのルートDNSサーバへのクエリ数を調べ，第4.3節にて筑波大学のフルリゾルバからルートDNSサーバへのクエリ数を確認し，第4.4節にて一般的なフルリゾルバソフトウェアに筑波大学でのクライアントクエリを再生してルートDNSサーバへのクエリ数を調査する．その後，第4.5節にて，ルートDNSサーバへのクエリ数の削減方法を提案し，シミュレーション及び実装により，評価を行なう．

表 4.1: DNS-OARC ルートデータセット

年	開始 (UTC)	終了 (UTC)	ルート DNS サーバの リスト	クエリ数 $*10^9$	送信元 IP アドレス数 $*10^6$
2011	Apr 12 1200	Apr 14 1200	a,c,d,e,f,h,j, k,l,m (10/13)	26.1	7.59
2012	Apr 17 1200	Apr 19 1200	a,c,e,f,h,j, k,l,m (9/13)	26.2	8.99
2013	May 28 1200	May 30 1200	a,c,d,e,f,h,j, k,l,m (10/13)	27.8	8.54

4.2 ルート DNS サーバでのクエリ解析

DNS-OARC は，2006 年から毎年 48 時間，ルート DNS サーバのパケットキャプチャを実施し，収集したデータを研究者に提供している．そのうち 2011 年から 2013 年の 3 年分のデータを評価し，IP アドレスごとのルート DNS サーバへのクエリ数を調査した [1] ．毎年取得できているルート DNS サーバが変わることと，各サーバでのすべてのデータを取得できているわけではないことから単純に比較することはできないが，2013 年であれば 854 万アドレスからの 278 億クエリを評価できた．評価したルートデータセットの概要を表 4.1 に，クエリ数の多いものから順に並べた IP アドレスごとのクエリ数を図 4.1 に示す．図 4.1 の横軸は IP アドレスの順位であり，縦軸はその順位の IP アドレスからのクエリ数である．

図 4.1 によると，最多クエリを出すアドレスは 48 時間で 8000 万以上のクエリを送っている．また，48 時間で 10 万クエリ以上送っているアドレ

[1] DNS-OARC as the data source of Root dataset.

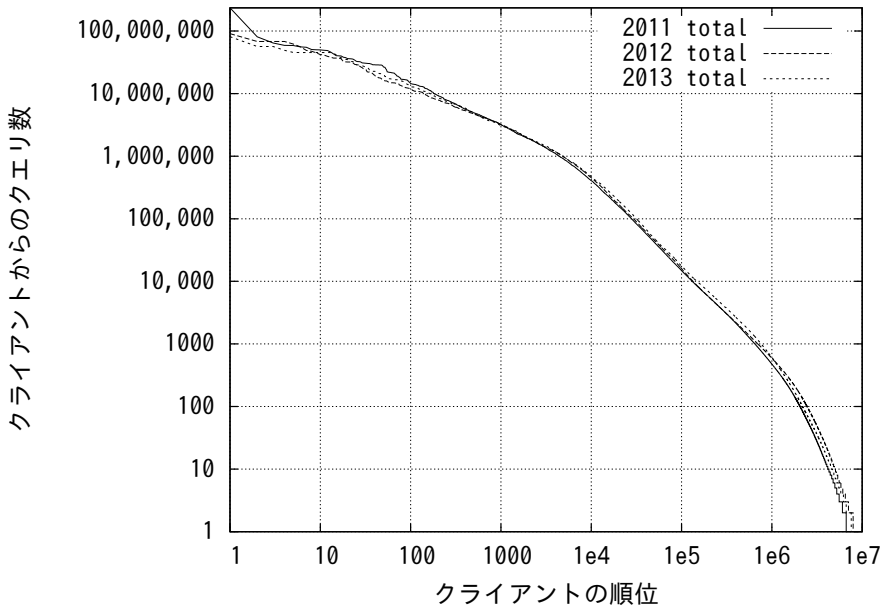

図 4.1: クエリ数が多いものから順に並べた個々の送信元 IP アドレスからルート DNS サーバへの 48 時間のクエリ数

表 4.2: 既知のアドレスからルート DNS サーバへの 48 時間のクエリ数 (2012 年)

IP address	ルート DNS サーバへのクエリ			
	総数	不在 TLD	“.”	存在 TLD
大学の フルリゾルバ	109,215	12,200	1,298	95,717

スが 3 万以上ある．DNS キャッシュの仕組みを考えると，この数は大き過ぎ，なんらかの異常があると考えられる．

この件に関連して著者の所属する大学のフルリゾルバについて調査した結果を表 4.2 に示す．このフルリゾルバは，BIND 9.6 をフルリゾルバ機能のみ使用する設定で，大学の内部からのアクセスだけ許して使用するという，推奨されている標準的な使用法であったが，ルート DNS サーバへのクエリ総数は予測していたものより多く，48 時間で 10 万以上であった．図 4.1 によると，ルート DNS サーバへのクエリ総数 109,215 は上位約 3 万位で，ルート DNS サーバへのクエリがキャッシュの仕組みを前提に予測していた数より多い現象の分析に好適のフルリゾルバである事がわかった．

4.3 フルリゾルバでのクエリ解析

上記分析の結果を考慮し，筆者らは筑波大学情報環境機構学術情報メディアセンターが運用する筑波大学全学向けフルリゾルバ (2 台運用しているうちの稼動負荷の大きなもの 1 台) で DNS クエリを収集し，その挙動を分析した．データ収集は実運用しているサーバの設定変更を避けて

タッピングにより実施した．タッピング点ではフルリゾルバの入出力パケットを全て取得し，キャプチャデータとして，スタブリゾルバとフルリゾルバの間と，フルリゾルバから権威 DNS サーバの間のパケットを取得した．計測したフルリゾルバは，学生寮や学生向け Wi-Fi ネットワークを含む大学全体からアクセスされるため，中規模の ISP が運用するフルリゾルバに似た利用傾向であると考えられる．

DNS クエリ情報にはエンドユーザの情報が含まれるため，1 台のマシンにデータを蓄積し，データを持ち出さず，その中で評価した．また，そのマシンには小人数の研究者のみアクセス可能とし，そのマシンからは統計情報と，クエリ再生実験向けにクエリ送信元 IP アドレスを除いたクエリ名とタイミング情報だけを取り出した．

ルート DNS サーバでの 2012 年のデータ取得期間と同じ 48 時間の筑波大学でのキャプチャデータの概要を表 4.3 に示す．ルート DNS サーバへは 118,360 のクエリを送っており，表 4.2 での 109,215 とは異なる．これはルートデータセットが一部のルート DNS サーバのデータのみで，データ収集の対象でないルート DNS サーバが存在するためである．

表 4.3 のルート DNS サーバへのクエリのうち存在応答のものは，105,781 と非常に多く異常である．2012 年 4 月 19 日の委任された TLD 数は 313 [21] であったことと，ルートゾーンの委任情報の TTL 値は 2 日であったため，キャッシュが理想的に動作した場合，48 時間で最大で 320 程度のクエリ数となると考えられる．また，キャッシュに存在しない TLD の外部名ネームサーバホスト名のクエリを複数同時に開始する場合や，A と AAAA の名前解決を同時に開始する場合，1 つの TLD 名に関して複数回クエリが送信され，存在する 313 より大きな数のクエリがルートに到達すると考えられるが，それを加味したとしても 10 万を越える数の説明はつかない．

存在しない TLD を検索するクエリを以下では不在クエリ，応答を不在

表 4.3: 筑波大学でのキャプチャデータ

データ取得開始 (UTC)	2012/4/17	12:00
データ取得完了 (UTC)	2012/4/19	12:00
取得時間	48 時間	
総パケット数	72,355,778	418 pps
クライアントからのクエリ数	28,815,955	166 qps
存在しない TLD のクエリ数	1,026,487	3.56%
クエリ元 IP アドレス数	8,429	
クエリ名数	551,226	
存在しない TLD のクエリ名数	8,459	
クエリ名中の存在する TLD	229	
クエリタイプ A	55.6%	
クエリタイプ AAAA	22.1%	
クエリタイプ PTR	21.5%	
クエリタイプ その他	0.7%	
権威 DNS サーバへのクエリ数	7,499,961	
権威 DNS サーバからの応答数	7,329,795	
ルート DNS サーバからの応答数	118,360	
ルート DNS サーバからの応答のうち存在応答のもの	105,781	89.4%
ルート DNS サーバからの応答のうち不存在応答のもの	12,579	10.6%
TLD DNS サーバからの応答数	687,365	

応答とする．不在応答についてはクエリ名が完全一致するものについてキャッシュが有効になることと，存在応答と異なり，キャッシュ可能時間を 1 時間から 3 時間とすることが推奨されているため [3]，キャッシュ効率が低下する．筑波大学のデータの場合，表 4.3 の存在しない TLD のクエリ数 1,026,487 (クライアントからの全クエリの 3.56%) からルート DNS サーバの不在応答となるクエリが 12,579 (ルート DNS サーバへのクエリ数の 10.6%) 生成され，98.8%キャッシュヒットしている．キャッシュヒット率が高いのは，連続的にクエリされる存在しない TLD のドメイン名が多いためである．また，存在しない TLD のクエリは 10.6%のルート DNS サーバへのクエリを生成しており，ルート DNS サーバへの影響が大きい．

4.4 クエリ再生実験による詳細分析

ここまでの測定の対象としたフルリゾルバは，2012 年当時にはサポートされていた BIND 9.6 系列のバージョンで，フルリゾルバのみ動作させ，特殊な設定を行わない標準的な構成であった．4.3 節で示した通り，測定対象のフルリゾルバが存在する TLD に関するクエリをルート DNS サーバに 48 時間で 10 万以上送ることは異常だと考えられる．

この現象を詳細に分析するために，対象期間のキャプチャデータを使いてクライアントクエリを再生し，再生されたクライアントクエリを受け取ったフルリゾルバが権威 DNS サーバへ送るパケットを分析した．図 4.2 にこのクエリ再生実験環境を示す．すべてのクライアントからのクエリを 1 つのアドレスからフルリゾルバに送り，すべての入出力パケットをキャプチャし，評価する構成である．2014 年 2 月時点で最新の BIND 9 と Unbound を複数の設定で評価した．

BIND 9[27] は，Internet Systems Consortium, Inc が開発している DNS サーバソフトウェアで権威 DNS サーバ，フルリゾルバ，スタブリゾルバの

すべての機能を持ち，リリース時点で標準化済の仕様をすべて実装する．BIND 9 は多くの OS で採用されており，多くの組織で使用されている．Unbound[37] は NLnet Labs が開発しているフルリゾルバで，DNSSEC 検証を含む機能をすべて実装している．Unbound は 2013 年からいくつかの OS で採用され，今後普及が進むと見込まれる．DNSSEC まで対応したフルリゾルバで無料のものは，2013 年の時点では BIND 9 と Unbound しかないため，この二つのソフトウェアを評価対象に選定した．

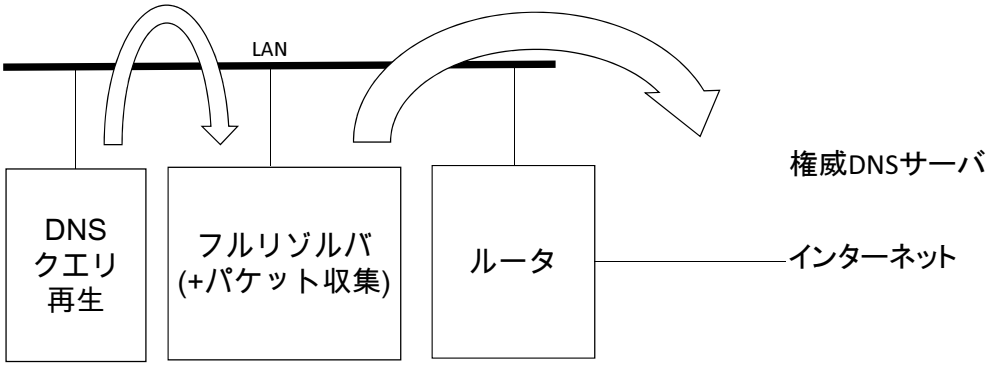

図 4.2: クエリ再生実験環境

評価結果を表 4.4 に示す．評価に用いた BIND 9 バージョンは 9.9.5 であり，Unbound は 1.4.21 である．/D は DNSSEC 検証を行なった場合を示し，/L はキャッシュ領域を大きくとった設定を示す．具体的には，msg-cache-size: 1024m，rrset-cache-size: 1024m，infra-cache-numhosts: 500000 という 3 行を unbound.conf に記述した．msg-cache-size は DNS メッセージ全体を保持するキャッシュ(msg_cache) の容量を指定するオプションであり，標準は 4 メガバイト (MB) である．rrset-cache-size は RRSet を保持するキャッシュ(rrset_cache) の容量を指定するオプションで，標準は 4MB である．infra-cache-numhosts は DNS の名前解決中にアクセスする権威 DNS サーバの IP アドレスごとの状態を保持しておくキャッシュ(infra_cache) のサイズを指定するオプションで，標準は 10000 である．BIND 9 の場合，

表 4.4: クエリ再生によるフルリゾルバから権威 DNS サーバへのクエリ数

サーバ	root	TLD	その他
実測値	118,360	687,365	6,524,070
BIND 9	163,187	842,592	7,377,712
BIND 9/D	663,647	1,061,886	7,235,743
Unbound	100,223	3,916,313	16,048,069
Unbound/L	13,300	870,650	9,102,884
Unbound/L/D	12,662	1,423,789	9,112,902

標準のキャッシュサイズは無制限である．今回のテスト環境では 3072MB のメモリを用意したため，Unbound でキャッシュ領域を増やした設定と同様の上限であると考えられる．

表 4.4 より，BIND 9，Unbound ともに標準的な設定では 48 時間でルート DNS サーバに 10 万以上クエリを送信していることがわかる．また，Unbound では，キャッシュ領域を大きくとると，ルート DNS サーバへのクエリを 1/7 以下に，TLD へのクエリを 1/4 以下に，他の権威 DNS サーバへのクエリを 6 割以下に減らせていることがわかる．これは，標準のキャッシュ領域が小さく，キャッシュサイズを大きく越える情報を扱ったため，性能が劣化したと考えられる．Unbound の実装者に標準のキャッシュサイズが小さいことを指摘したところ，標準の設定値は組み込み機器などのメモリが少ない場合や，一般家庭や SOHO などの小規模なところを対象としていて，大学のような大規模なところではチューニングしてキャッシュサイズを増やすようにとのことであった．

Unbound では，名前解決中に msg_cache と rrset_cache を用途に応じて使うこと，今回のテストデータではクエリ名数が 551,226 であり，それぞ

れ複数のタイプの値を扱うこと，DNS 応答のパケットサイズは 100 バイト程度以上であることより，$551,226 * 2 * 100$ バイト (約 105MB) 以上のキャッシュメモリを使用する．標準の 4MB で不足することは明らかで，msg-cache-size と rrset-cache-size を 1024MB にする必要があった．また，DNS サーバのアドレス数については 14 万以上であったため，infra-cache-numhosts を増やすことも必要であった．

BIND 9 で DNSSEC 検証を有効にすると，ルート DNS サーバへのクエリが 4 倍程度に増えることがわかる．TLD が DNSSEC 対応している場合，ルート DNS サーバからの委任応答に DS RR が含まれるが，BIND 9 の場合はその DS RR を無視して別途 DS RR の取得を行なうため，大きな値になったと考えられる．

Unbound の場合は DNSSEC 検証の有無でルート DNS サーバへのクエリはほとんど変化せず，表 4.4 では逆に減っている．Unbound の場合は，DNSSEC 検証の有効化でルート DNS サーバへ DNSKEY クエリを 1 日 1 回以上送るが，委任応答に含まれる DS RR を有効に使い，委任応答に続くタイプ DS のクエリを送らないため，DNSSEC 検証の有無での差が少ない．

次にルート DNS サーバへのクエリを詳細に分析した．まずルート DNS サーバへのクエリについて，キャッシュが最大限動作した場合に，キャッシュすることで削減可能なクエリ数を求めた．同じ TLD や存在しないドメイン名のクエリがルート DNS サーバに連続して送られることがある．これは，クライアントが A と AAAA を同時に問い合わせたり，フルリゾルバが外部名ネームサーバホスト名の名前解決をする場合に名前解決に要する時間を最小化するために複数のネームサーバホスト名のタイプ A とタイプ AAAA の名前解決を同時に開始するためである．それらのクエリは検索の速度向上のためであり，削除不可能であるため，一秒以内に連続する存在する TLD へのクエリと，存在しない TLD を含むクエリ名

表 4.5: クエリ再生によるルート DNS サーバへのクエリの詳細

サーバ	総数	不在	不在で削減可能	存在	存在で削減可能
実測値	118,360	12,579 10.6%	521 0.4%	105,781 89.4%	105,169 88.9%
BIND 9	163,187	12,975 8.0%	764 0.5%	150,212 92.0%	149,374 91.5%
BIND/D	663,647	12,727 1.9%	743 0.1%	650,920 98.1%	649,369 97.8%
Unbound	100,223	25,914 25.9%	3,563 3.6%	74,309 74.1%	73,480 73.3%
Unbound/L	13,300	11,444 86.0%	634 4.8%	1,856 14.0%	375 2.8%
Unbound/L/D	12,662	11,140 88.0%	524 4.1%	1,522 12.0%	645 5.0%

を削減できないとした．以下，存在する TLD については TLD が一致するクエリ名，存在しない TLD についてはクエリ名が完全一致するクエリ名を一致判定するクエリ名として分析した．削減可能なルート DNS サーバへのクエリは，クエリ名ごとに，キャッシュ1 秒後からキャッシュ有効期間までにルート DNS サーバに到達した一致判定するクエリ名のクエリである．分析結果を表 4.5 に示す．本分析では”.” クエリを存在する TLD クエリに含めた．

筑波大学のフルリゾルバでの実測値で，存在しない TLD のクエリのもので削減可能なものは 521 だけであり，キャッシュサイズは十分であると考えられる．また，存在する TLD についてのルート DNS サーバへのク

エリでキャッシュ可能なものは 105,169 であり，存在する TLD についてのルート DNS サーバへのクエリを 99.4%削減可能である．

評価結果によると，ルート DNS サーバへの不在応答となるクエリ数は，Unbound 標準の場合を除き，どの場合でも 11,000 から 13,000 の間であった．元データより，クライアントからのクエリのうち存在しない TLD についてのクエリ名数は 8,459 であったため，定期的なクエリを考慮すると，存在しない TLD に関するクエリについてはキャッシュが有効に動作していた．

なお，BIND 9 と Unbound でネガティブキャッシュの TTL 値の扱いが異なり，BIND 9 は標準で 3 時間を上限としてキャッシュするが，Unbound は 24 時間を上限としてキャッシュする．そのため，存在しない TLD の情報については Unbound の方がキャッシュでの生存期間が長いため，BIND 9 に比べて Unbound の方が 1000 以上存在しない TLD に関するクエリが少なくなっていた．

存在する TLD のクエリに関して，BIND 9 では削除可能なクエリが非常に多い．それらを詳細に分析したところ，ネームサーバホスト名のクエリが多いことがわかった．また，それらが送られるタイミングは委任情報に外部名ネームサーバホスト名が含まれていた場合だけではなく，名前解決時に付加的に権威 DNS サーバから添付されるゾーンのネームサーバホスト情報を受けとった場合に，ルート DNS サーバから外部名ネームサーバホスト名の名前解決を行なうことが多かった．ルート DNS サーバへのクエリ数が多い原因は，BIND 9 の実装がそのような状況を十分考慮していないためと考えられる．

Unbound でキャッシュサイズを増やした場合のルート DNS サーバへのクエリ数は理想値に近く，ルート DNS サーバへのクエリのうち存在する TLD のクエリで削減可能なものは 375 (存在する TLD クエリ総数の 20.2%) であった．Unbound が削減可能なルート DNS サーバへのクエリ

を送る場合の動作を調べたところ，BIND 9 と同様にネームサーバホスト名のクエリが多く，それらが送られるタイミングは外部名のネームサーバホスト名の名前解決だけではなく，名前解決時に付加的に権威 DNS サーバから添付されるゾーンのネームサーバホスト情報を与えられたときが多かった．BIND 9 も Unbound も名前解決時に付加的に権威 DNS サーバから添付されるゾーンのネームサーバホスト情報の扱いが効率的でない．この点の対応策などを次節で検討する．

また，キャッシュサイズを増やした場合の Unbound では，全クライアントクエリの 3.56%の存在しない TLD に関するクエリがルート DNS サーバへのクエリの 86.0%を生成しており，存在しない TLD についてのクエリの削減が重要である．

4.5 ルート DNS サーバへのクエリ削減の提案と評価

4.4 節で示した無駄なルート DNS サーバへのクエリを削減する方法として以下の 3 案を提案する．

1. フルリゾルバとして Unbound を使用し，キャッシュサイズを増やす
2. 名前解決とキャッシュアルゴリズムを改善する
3. NSEC リソースレコードにより，存在しない TLD クエリを削除する

第 1 案は 4.4 節で述べた通りであるので，以下第 2 案，第 3 案について述べる．

4.5.1 名前解決とキャッシュアルゴリズムの改善

BIND 9 と Unbound の両方で，名前解決時に付加的に権威 DNS サーバから添付されるゾーンのネームサーバホスト情報を与えられた場合にルー

トDNSサーバに無駄なクエリを送る傾向が強い．これはRFC 2181 [22]の問題と考えられる．

RFC 2181 Section 5.4.1 Ranking(ランキング)ではDNSサーバが扱うデータの信頼度を以下に示す順序で定義している．

(1) グルーを除くプライマリゾーンからのデータ

(2) グルーを除くゾーン転送で得たデータ

(3) 管理権限のある応答中の応答セクションにある管理権限のあるデータ (問い合わせたクエリに対応する応答)

(4) 管理権限のある応答中の管理権限セクションのデータ

(5) プライマリゾーンかゾーン転送で得たグルー

(6) (6a) 管理権限のない応答中の応答セクションのデータと (6b) 管理権限のある応答中の応答セクションの管理権限のないデータ

(7) (7a) 管理権限のある応答中の付加データ，(7b) 管理権限のない応答中の管理権限セクションのデータ (委任情報)，(7c) 管理権限のない応答中の付加情報

(1),(2),(5) のプライマリゾーンやゾーン転送は権威DNSサーバとして動作させた場合のデータの入手法であり，管理者が設定したものであるため，優先度が高い．プライマリゾーンはゾーン情報を指定したファイルなどであり，ゾーン転送はゾーン情報を複製する仕組みである．フルリゾルバとして動作させた場合には，名前解決の前に固定設定のものを優先するという意味となる．

管理権限のある応答とは，親ゾーンから委任されたDNSサーバからの応答で，Authoritative Answer(AA) ビットがセットされているものであ

り，委任情報とグルーを含まない．応答には，応答セクション，管理権限セクション，付加情報セクションの 3 つの部分があり，クエリに対する応答は応答セクションに入る．管理権限のあるデータとは，応答中のリソースレコードのドメイン名が，クエリを送った権威 DNS サーバに委任されたドメイン名と同じか，委任されたドメイン名の子孫のドメイン名であるデータである．

上記ランキングには，次の二つの問題がある．

問題 1: フルリゾルバがスタブリゾルバに管理権限のないデータ (5,6,7) や，直接問い合わせたデータではない添付されたデータ (4) を応答する．

問題 2: 名前管理の管理権限を委任する委任情報 (7b) を子ゾーンの情報 (3 や 4) で上書きすることは，動作を複雑にするだけでなく，親ゾーンから子ゾーンに委任する管理権限の考え方に矛盾する．

DNS の仕様上の問題として，委任の子側では委任情報と同じ情報を委任されたゾーン内に書く必要がある事から派生する問題がある．すなわち委任情報とゾーン内の管理権限のある情報が異なると問題 2 の影響を受ける．具体的には，TLD に登録したドメイン名の委任情報が，ドメイン名登録者が運用する子側のネームサーバホスト情報で上書きされ (委任情報 (7b) を使った名前解決中にランキング (3) や (4) の応答を受けとる)，その後の名前解決の動作が複雑になる．

この複雑さによるバグが何度か報告されている．例えば，子ゾーン側の管理権限のある NS RR の TTL 値を親ゾーンの TTL 値と比べて非常に長くしておくと，TLD からドメイン名が消えた後もそのドメイン名の情報がキャッシュに存在し続け，名前解決を無効にできない場合があるという問題があった．また，ゾーンの NS RRSet のキャッシュ保持期間が過ぎる時に，現在キャッシュされている NS RRSet の示す権威 DNS サーバに

問い合わせを行なう実装があり，幽霊ドメイン名というバグを報告された [29].

以上の問題を解決するため，以下のようなフルリゾルバを提案する．

a. 委任情報キャッシュと管理権限のある情報のキャッシュ(応答キャッシュ)を分離する．

b. 静的に指定された情報を最優先とし，そのうちの委任情報とグルーを委任情報キャッシュに，管理権限のある情報を応答キャッシュに初期値として蓄積する．

c. 管理権限のある応答までは委任情報とグルーを用いて名前解決をおこなう．委任情報とグルーを委任情報キャッシュに蓄積する．

d. 管理権限のある応答中のクエリに対応する応答以外の情報は廃棄し，管理権限のある応答を応答キャッシュに蓄積する．CNAME による別名の添付情報も廃棄する．

e. クライアントには管理権限のある応答のみ応答キャッシュを使用して応答する．

f. 外部名ネームサーバホスト名のアドレスはクライアントクエリと同様に名前解決し，管理権限のある情報として扱い，委任情報から参照する．

提案手法の a, d, e により，管理権限のない情報を応答する問題 1 を解決し，a, c, f により，委任情報の上書きを防ぎ，問題 2 を解決する．すなわち

- a により，委任情報と管理権限のある情報を分離する．
- b により，外部の権威 DNS サーバの情報に依存しない，固定設定情報を実現する．

- c により，委任情報だけによる名前解決を実施する．
- d により，管理権限のない情報をスタブリゾルバに応答することを防ぐ．
- f により，外部名ネームサーバホスト名の名前解決を，安全な形で実装する．

提案手法は，すべてのスタブリゾルバからのクエリに対して，委任情報が指定する権威 DNS サーバが保持する管理権限のある情報を応答するため，妥当である．また著名な参考書 [42] に書かれている DNS の動作では，フルリゾルバはルート DNS サーバから順に名前空間を検索することとなっており，グルーを含む委任情報のみを使用して名前解決を行なうという説明が多い．提案手法は一般的な解説通りの動作となる．

提案手法には，委任情報の NS RR の TTL 値とゾーン側の NS RR の TTL 値に異なる値を指定し，ゾーン側の TTL 値を優先させ，TTL 値を小さくするという制御が不可能となる問題点がある．ゾーン側で小さな TTL 値を設定することは実運用でよく行われている．しかし，そもそも委任情報とゾーン内に同じキーで TTL 値を含む別の値を設定可能な点が DNS の設計ミスであり，そのような制御を行なうべきではない．NS RR の TTL 値を小さくし，キャッシュ時間を短く制御したい場合として，DNS サーバの変更が考えられ，自由に設定できない委任情報の TTL 値を小さくするためにゾーン内の権威ある NS RR の TTL 値を小さくし，フルリゾルバでキャッシュする NS RR の TTL 値を小さくする事を試みる事がある．しかし，この手法は委任情報からゾーン内の NS RR への置き換えか必ず行なわれるわけではないため確実ではない．また，この手法の弊害として，TLD で TTL 値を 1 日以上としてキャッシュ保持時間を長くしている場合でも，登録ドメイン名で TTL 値を小さくすると，1 日以上の TTL 値を意図している TLD の DNS サーバへのクエリが増える原因にな

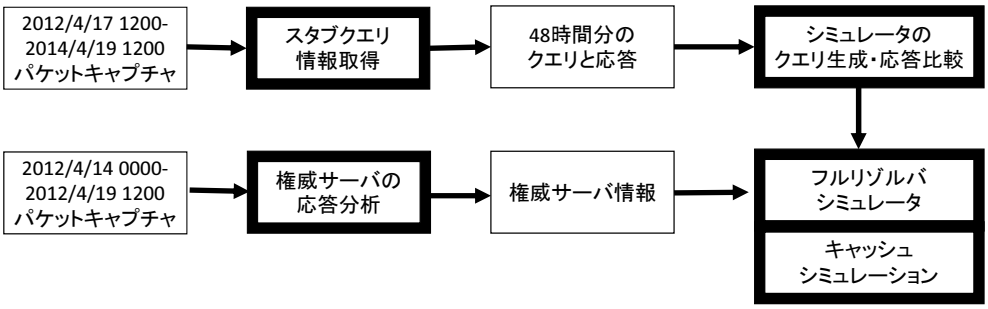

図 4.3: フルリゾルバシミュレータの構成

る．TTL 値を制御しない DNS サーバの変更手順を採用すべきである．

提案手法の問題がないこと，ルート DNS サーバなどへのクエリ数を削減可能であることを示すために，フルリゾルバでのパケットキャプチャをもとに，フルリゾルバのシミュレータを試作し，クライアントからのクエリを与え，権威 DNS サーバへのクエリ数を評価した．図 4.3 にシミュレータの構成を示す．

シミュレーションを行なう過程で，CDN などで頻繁にアドレスが変化するアドレスや異常応答のため，過去のパケットキャプチャデータだけでは応答の完全一致を求めることは困難であることがわかった．CDN の場合は複数の応答をすべて記録しておき，可能性があるアドレスを列挙することで，アドレスの一致を確認した．また表 4.3 より，99.3%のクエリが A,AAAA,PTR を対象としていること，エンドユーザのウェブアクセスでは A,AAAA を用いることから応答の比較は A,AAAA,PTR の 3 タイプでのみ行なった．

4.3 節で使用した 2012 年 4 月 17 日からのキャプチャデータを用いて，フルリゾルバのシミュレーションを行ない，提案したアルゴリズムで応答が一致することと，ルート DNS サーバなどの権威 DNS サーバへのクエリ数を評価した．評価結果のうち，応答の状態を表 4.6 に，権威 DNS

表 4.6: フルリゾルバシミュレーションでの応答の状態

クエリ数	28,359,467
一致した応答	28,161,142 (99.3%)
比較していないタイプ (A,AAAA,PTR 以外)	197,950 (0.7%)
不一致	375 (0.001%)

表 4.7: フルリゾルバシミュレーションでの権威 DNS サーバへのクエリ数

	ルート DNS サーバ			TLD DNS サーバ	その他の権威 DNS サーバ
		不在	存在		
実測値	118,360	12,579 10.6%	105,781 89.4%	687,365	6,524,070
Unbound/L	13,300	11,444 86.0%	1,856 14.0%	870,650	9,102,884
シミュレーション	13,257	12,234 92.3%	1,023 7.7%	350,873	5,542,808

サーバへのクエリ数を表 4.7 に示す.

表 4.6 より，0.001%の不一致と比較していないタイプ 0.7%を除いた 99.3%の応答で，実際のフルリゾルバの応答とシミュレータの応答が一致した．一致しなかった 0.001%はパケットキャプチャの失敗か，シミュレータが BIND 9 の動作を完全に模擬できていない場合である．具体的には，試作したシミュレータは権威 DNS サーバの設定変更中で親ゾーンの NS 設定と子ゾーンの NS 設定が違っている場合の扱いが BIND 9 と異っており，それが不一致の原因となっている．ルート DNS サーバへのクエリ数検討の目的に照せば問題となる違いではなく，そもそも我々の提案

は親ゾーンの設定を子ゾーンの設定で上書きさせない提案であり，この違いは当然である．

表 4.6 より，ルート DNS サーバへの存在しない TLD のクエリ数は，筑波大学のフルリゾルバの実測値や Unbound でのクエリ再生実験結果とほぼ一致する結果が得られた．存在する TLD のクエリ数については 1,023 と，Unbound での測定値の 55%となり，実装の工夫で減らせることがわかった．TLD DNS サーバへのクエリ数は実測値の 51%となり，減らせた．その他の権威 DNS サーバへのクエリ数は実測値の 85%で，クエリに対する直接の応答以外の情報を廃棄したことによるクエリ数の増大は起きないことがわかった．

4.5.2 NSEC リソースレコードによる存在しない TLD クエリの削除

4.3 節で示した通り，48 時間でのクライアントからの 28,815,955 クエリのうち，1,026,487 が存在しない TLD へのクエリであり，Unbound を使った場合や，理想的なフルリゾルバの場合でもルート DNS サーバへ 11,000 以上の存在しない TLD のクエリを送ることがわかった．そのクエリを減らす方法として，DNSSEC で追加された NSEC リソースレコードを用いる方法を提案する．

DNSSEC ではドメイン名が存在しないことを示すために，存在する名前を辞書順に並べた範囲情報に署名し，その範囲内のクエリに対して範囲を返す．具体的には，2014 年 7 月 23 日現在，com TLD と community TLD の間にはラベルが存在しないため，ルート DNS サーバに com1 を問い合わせると以下の NSEC リソースレコードが返される．

```
com. IN NSEC community. NS DS RRSIG NSEC
```

この情報はDNSSEC検証を行なうフルリゾルバにキャッシュされ，さらにcom1が存在しないことがキャッシュされる．このときにcom2やexample.com1のクエリを受け取った場合，comのNSEC RRが示す範囲にあるため，即時にエラーを返すことが可能であるが，実際にはNSEC RRは使用されず，ルートDNSサーバにcom2やexample.com1を問い合わせる．これはRFC 4035 Section 4.5. Response Cachingに，新しく登録されたラベルが即時に使用できるように，フルリゾルバはNSEC RRを用いて不存在情報を合成すべきではないと記述されているためである．

NSECを積極的に使う改善により，ルートDNSサーバへのクエリを削除することを考える．ルートのみに限定することは容易で，NSECに対応するRRSIGのSignerフィールドがルート (“.”) の場合だけその結果を用いるようにすればよい．

ルートゾーンのネガティブキャッシュTTLの値は86400であるが，現在，ルートゾーンへの新規登録は新gTLDプログラムによるものがほとんどであり，新gTLDがルートゾーンに追加されたあと90日はnic.TLD以外の登録をしないという規則があるため，事実上，新gTLDの使用が86,400秒 (1日) 遅くなっても問題はない．

そこでUnbound 1.4.21を改造してNSECを積極的に使い，ルートDNSサーバへの存在しないTLDクエリを削減する効果を計測した．UnboundにはDNSSEC Lookaside Validator (DLV, [57]) を効率化するコードのなかにNSECを積極的に使い，DLVクエリを削減する機能があったため，それを流用し実装した．

NSECによる削減機能を追加したUnboundにクエリ再生を行ない，パケット数を調べた結果を表4.8に示し，筑波大学のフルリゾルバでの実測値，Unboundでの再生実験値，Unboundでの値をもとにした理論値，NSECによるクエリ削減の理論値と比較する．

表4.8によると，NSECによる削減では理論的には83.3%の不存在TLD

表 4.8: NSEC によるクエリ削減の評価結果

	権威 DNS サーバへのクエリ数				
	ルート DNS サーバ			TLD	その他
	総数	不在	存在		
実測値	118,360	12,579	105,781	687,365	6,524,070
Unbound/L/D	12,662	11,140	1,522	1,423,789	9,112,902
NSEC により削減できるクエリ数 (理論値)		9,282			
Unbound からの削減		83.3%			
NSEC による削減 (理論値)	3,380	1,858	1,522		
Unbound/L/D 比	26.7%	16.7%			
NSEC による削減結果	3,997	2,652	1,345	1,418,998	9,124,388
Unbound/L/D 比	31.6%	23.8%			
Unbound からの削減	68.4%	76.2%			
実測値 比	3.4%	21.1%		2.06 倍	1.40 倍
実測値からの削減	96.6%	78.9%			

クエリを削除可能であったが，Unbound にパッチをあてたものは 76.2%の削除となった．もともとの実測値と比較すると，不在クエリの 78.9%を削除でき，Unbound の性能と併せて，ルート DNS サーバへのクエリを 96.6%削減できた．DNSSEC 検証の影響で，TLD DNS サーバへのクエリは実測値の 2.06 倍に，その他の権威 DNS サーバへのクエリは 1.40 倍に増加しているが，これは DNSSEC の使用による増加で，DNSSEC 使用を前提とすれば，この改良による変化はない．

本章では，ルート DNS サーバへのクエリ数削減を目的としているが，NSEC リソースレコードを使用した不存在クエリ対策は，実在しない名前のクエリを利用した DoS 攻撃に対しても有効である．また，入力間違いによる存在しない名前検索に対しても，権威 DNS サーバへのクエリを減らすことができ，応答遅延時間を短縮できる可能性がある．

本手法は，NSEC 方式で DNSSEC 署名されたゾーンでのみ適用可能である．所有者名のハッシュで連鎖を作る NSEC3 [39] 方式で DNSSEC 署名されているゾーンについても同じ手法が適用可能であると考えられるが，実際には NSEC3 Opt-Out [39] 方式で DNSSEC 署名された TLD が多いため，効果を得られないと考えられる．NSEC3 Opt-Out 方式とは DNSSEC 対応ではない委任情報を除外する方式で，NSEC3 リソースレコードは DNSSEC 対応ではない委任が存在しないことを示さない．NSEC3 Opt-Out 方式は jp や com, net などの多くの TLD で採用されており，NSEC3 Opt-Out 方式への対応は今後の課題である．

4.6 結言

本章では，ルート DNS サーバに大量の本来不必要なクエリを出しているアドレスが大量にあることを示し，標準的な設定のフルリゾルバからも 48 時間で 10 万以上のクエリを送る場合があることを示した．

現象を詳細に解析した結果，世界で広く使われている BIND 9 はルート DNS サーバに無駄にクエリを送ること，存在しない TLD のクエリによるルート DNS サーバへのクエリも多いこと，その原因の一つが実装の不備によることを示した．

更に本章ではこのような状態を改善するための方法として，1) フルリゾルバの設定による改善法，2) フルリゾルバの名前解決とキャッシュアルゴリズムの改善法，3)NSEC リソースレコードを使った改善法，の 3 つの手法を示し，それぞれの効果を定量的に評価した．最終的には，変更した Unbound でルート DNS サーバへのクエリを 96.6% 減らせる事を示した．

BIND 9 がルート DNS サーバに大量にクエリを出す不具合は，DNS-OARC ワークショップにて問題点を指摘しており [23]，現在改良が検討されている．4.5.1 節で示したアルゴリズムと 4.5.2 節で示した NSEC リソースレコードによる不存在 TLD クエリの削減を組み合わせるとさらにルート DNS サーバへのクエリを削減可能であり，今後国際的なコミュニティーと協力しながら実装を行ないたい．

第5章　結論

インターネットは重要な社会インフラの一つとなり，インターネットで使用されるドメイン名システム (DNS) の重要性は増している．今後長期に渡り DNS を持続させるためには，実際の名前解決を行なうフルリゾルバと，名前解決の起点となるルート DNS サーバへの負荷の把握と低減が重要である．

この様な背景の中，第 3 章では，筑波大学のフルリゾルバのクエリ分析を行ない，キャッシュ効率増大や遅延低減のための適切な設定の提案を行なった．

(1) 筑波大学のフルリゾルバのクエリ分析

本研究では，まず近年の IPv6 や CDN の影響を評価するため，筑波大学情報環境機構学術情報メディアセンターの運用するフルリゾルバの入出力パケットを分析した．そのフルリゾルバは全学のユーザを対象としており，学生寮や Wi-Fi 接続の通常の PC が参照するため，利用帯域等のデータから中規模 ISP のフルリゾルバに近いと考えられる．時期によるが一ヶ月で 8800 から 11000 のクライアント IP アドレスを観測した．

キャプチャデータの分析の結果，IPv6 対応 OS の普及により IPv6 アドレス (AAAA) クエリが増加したため，クライアントからのクエリが 41%増加したこと，CDN などの DNS を使用した広域負荷分散サービスの普及により，小さな TTL 値，CNAME，外部名ネーム

サーバホスト名の使用が増えていることがわかった．結果として，エラー応答の割合の増加やキャッシュヒット率の低下，応答遅延の増大がみられた．

また，2012 年 6 月 6 日の World IPv6 Launch の前後で AAAA クエリに存在応答が戻る割合が 2.0%から 21.2%に増加した．これは，World IPv6 Launch 以降，よく使用されるドメイン名の IPv6 対応が進んだことを示している．

(2) キャッシュ効率増大や遅延低減のための適切な設定の提案

(1) の結果は，(a) 無用な AAAA クエリの低減，(b) 適切に大きな TTL 値の設定，(c)CNAME の削減，(d) 内部名ネームサーバホスト名の適切な設定，が重要な運用課題である事を示している．これらの結果を背景に，本研究では以下の指摘と提案を行った．

(a)IPv6 対応 OS(比較的新しいバージョンの Microsoft Windows や Apple Mac OS X，Linux など) は IPv6 接続性がない場合でも AAAA クエリをフルリゾルバに送信する．これら IPv6 対応 OS が，IPv6 接続性の不存在を検知し，不要な AAAA クエリを省略することで，フルリゾルバの負荷を軽減することができる．

(b)DNS を利用した広域負荷分散で使用される小さな TTL 値は，キャッシュヒット率を下げ，フルリゾルバの負荷を上げ，権威 DNS サーバへのクエリ数を増大させる．名前解決にかかる平均時間も，TTL 値 300 以下で 31.7 ミリ秒が，TTL 値が 300 を超える場合には 25.1 ミリ秒と，TTL 値が大きい方が短い．適切に大きな TTL 値を使用する事で名前解決に要する時間を短縮できる．

(c)CNAME を使用する複雑なドメイン名の名前解決に要する平均

時間は 30.4 ミリ秒であったが，CNAME を使用しないドメイン名の場合は 24.9 ミリ秒であった．CNAME の使用を最小限にすることで名前解決に要する時間を短縮できる．

(d) 外部名ネームサーバホスト名を使用するドメイン名の名前解決に要する時間は平均 33.1 ミリ秒であったが，すべて内部名ネームサーバホスト名を使用するドメイン名の場合は平均 22.5 ミリ秒であり，内部名ネームサーバホスト名を適切に使用することで名前解決の時間を短縮できる．

これらの施策を行なうことで，フルリゾルバの負荷を低減でき，名前解決に要する時間を短縮できる．

次に，第 4 章において，ルート DNS サーバ及び筑波大学のフルリゾルバでのクエリ分析を行ない，一般的なフルリゾルバがルート DNS サーバに本来不必要なクエリを大量に送信することを示し，ルート DNS サーバへのクエリ低減方法の提案を行なった．

(3) ルート DNS サーバ及び筑波大学のフルリゾルバのクエリ分析

ルート DNS サーバでは年に一度パケットキャプチャを実施し，研究者にデータを提供している．そのデータを分析し，多くの IP アドレスから大量の本来不必要なクエリがルート DNS サーバに送られ，余分な負荷が発生している事を示した．具体的には，3 万以上の IP アドレスが 48 時間に 10 万以上のクエリをルート DNS サーバに送っていた．筑波大学のフルリゾルバも本来不要なクエリを大量にルートに送っていたことがわかり，動作を確認したところ，一般的な設定がされた中規模以上のフルリゾルバ共通の振舞いである事がわかった．現象を詳細に解析した結果，世界で広く使われている BIND 9 はルート DNS サーバに無駄なクエリを送ること，存在しな

いTLDに対するクエリによるルートDNSサーバへのクエリも多いことがわかった．

(4) ルートDNSサーバへのクエリ低減方法の提案

ルートDNSサーバへの不要クエリが多い問題に対し，本研究では (a) フルリゾルバの設定による改善法，(b) フルリゾルバの名前解決とキャッシュアルゴリズムの改善法，(c)NSECリソースレコードを使った改善法，の3つの手法を示し，それぞれの効果を定量的に評価した．

(a) フルリゾルバの設定による改善法は，BIND 9とは異なるUnboundという実装を用い，キャッシュサイズを増やすことである．

(b) アルゴリズムの改善では，フルリゾルバでのデータの扱いを改善した理想的なフルリゾルバアルゴリズムを規定し，名前解決のシミュレーションを行ない，ルートDNSサーバへのクエリ数が (a) と同様となることを示した．この改善にはRFC 2181の変更を必要とする．

(c)NSECリソースレコードを使った改善法はDNSSECの不存在応答検証の仕組みを用い，ルートDNSサーバに到達する存在しないTLDのクエリを減らす手法であり，Unboundにパッチをあてて実装し，評価を行なった．結果として (a)(b) では削減できない存在しない名前についてのルートDNSサーバへのクエリを78%減らすことができた．この改善にはRFC 4035の小規模な変更を必要とする．

(a)(c) の組み合わせで，ルートDNSサーバへのクエリを1/29に減らすことができる．ルートDNSサーバへのクエリを減らすことで，

ユーザからのクエリの応答時間の短縮とフルリゾルバの負荷の低減も同時に達成される．

これらの研究成果により，ユーザからのクエリの応答時間の短縮，フルリゾルバの負荷の低減，ルート DNS サーバへのクエリの低減が可能となる．将来的に実装が広まれば，現在のインターネットを長期に渡って維持することが可能となる．本研究の成果がインターネットの維持と発展に少しでも役に立てば幸いである．

謝辞

本論文は，筆者が筑波大学大学院システム情報工学研究科リスク工学専攻博士後期課程に在籍中の研究成果をまとめたものである．

本研究を進めるにあたっては，多くの方々のご指導，ご協力，ご支援を賜った．ここに，お世話になった皆様への感謝の意を表す．

本研究を進めるにあたり同専攻 吉田健一教授には指導教員として，終始有益なご指導とご助言，温かい励ましのお言葉を頂いた．ここに深謝の意を表する．

本研究を遂行するにあたり，同専攻 津田和彦教授，同専攻 倉橋節也准教授，同専攻 片岸一起准教授，並びに長岡技術科学大学 山崎克之教授には副査としてご助言，ご指導を頂いた．ここに深謝の意を表する．

また，発表会や論文投稿どで，様々なアドバイスやコメントを頂いた同専攻教員の方々，学会関係者に深く感謝の意を表する．

吉田研究室の各位には日頃より有益なご討論ご意見を頂いた．ここに感謝の意を表する．

筑波大学情報環境機構学術情報メディアセンター (システム情報系情報工学域) 佐藤聡准教授には学術情報メディアセンターが運用する筑波大学全学向けフルリゾルバのデータを収集し，解析する機会を与えていただくとともに有益なご指導ご助言を頂いた．ここに感謝の意を表する．

ルートデータセットを解析する機会を与えて頂いた DNS-OARC に敬意と感謝の意を表する．

更に，jp TLD DNS サーバのデータ提供並びに研究の支援を頂いた株

式会社 日本レジストリサービスに敬意と感謝の意を表する．特に佐野晋代表取締役副社長並びに三田村健史 技術本部長に深く感謝する．

参考文献

[1] J. Abley and W. Maton. AS112 Nameserver Operations. RFC 6304 (Informational), July 2011.

[2] Bernhard Ager, Wolfgang Mühlbauer, Georgios Smaragdakis, and Steve Uhlig. Comparing dns resolvers in the wild. *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pages 15–21, 2010.

[3] M. Andrews. Negative Caching of DNS Queries (DNS NCACHE). RFC 2308 (Proposed Standard), March 1998. Updated by RFCs 4035, 4033, 4034, 6604.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS Security Introduction and Requirements. RFC 4033 (Proposed Standard), March 2005. Updated by RFCs 6014, 6840.

[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014, 6840.

[6] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014, 6840, 6944.

[7] T. Asaba, k. claffy, O. Nakamura, and J. Murai. An analysis of international academic research network traffic between japan and other nations. *International Networking Conference (INET)*, Jan 1992.

[8] D. Atkins and R. Austein. Threat Analysis of the Domain Name System (DNS). RFC 3833 (Informational), August 2004.

[9] RJ Atkinson, SN Bhatti, and S. Rose. DNS Resource Records for the Identifier-Locator Network Protocol (ILNP). RFC 6742 (Experimental), November 2012.

[10] Saleem Bhatti and Randall Atkinson. Reducing dns caching. *14th IEEE Global Internet Symposium (GI2011)*, Apr 2011.

[11] A. Broido, H. Shang, M. Fomenkov, Y. Hyun, and k. claffy. The Windows of Private DNS Updates. *ACM SIGCOMM Computer Communication Review (CCR)*, 36(3):93–98, Jul 2006.

[12] N. Brownlee, k. claffy, and E. Nemeth. DNS Measurements at a Root Server. *IEEE Global Telecommunications Conference (GLOBECOM)*, Nov 2001.

[13] N. Brownlee, k. claffy, and E. Nemeth. DNS Root/gTLD Performance Measurements. *Usenix LISA*, Dec 2001.

[14] N. Brownlee and I. Ziedins. Response time distributions for global name servers. *Passive and Active Network Measurement Workshop (PAM)*, Mar 2002.

[15] S. Castro, D. Wessels, M. Fomenkov, and k. claffy. A Day at the Root of the Internet. *ACM SIGCOMM Computer Communication Review (CCR)*, 38(5):41–46, Oct 2008.

[16] S. Castro, M. Zhang, W. John, D. Wessels, and K. Claffy. Understanding and preparing for DNS evolution. *Traffic Monitoring and Analysis*, pages 1–16, 2010.

[17] k. claffy. A Day in the Life of the Internet: Proposed community-wide experiment. *ACM SIGCOMM Computer Communication Review (CCR)*, 36(5):39–40, Oct 2006.

[18] Carlo Contavalli, Wilmer van der Gaast, Sean Leach, and Darryl Rodden. Client subnet in DNS requests. Jan. 2011. draft-vandergaast-edns-client-subnet-00.txt.

[19] J. Crowcroft and I. Wakeman. Traffic analysis of some uk-us academic network data (1991). *Proceedings of INET'91*, 1991.

[20] Peter Danzig, Katia Obraczka, and Anant Kumar. An analysis of wide-area name server traffic: a study of the internet domain name system. *Proceeding of the 1992 ACM Conference on SIGCOMM*, pages 281–292, 1992.

[21] DNS-OARC. Root Zone Archive. https://www.dns-oarc.net/oarc/data/zfr/root.

[22] R. Elz and R. Bush. Clarifications to the DNS Specification. RFC 2181 (Proposed Standard), July 1997. Updated by RFCs 4035, 2535, 4343, 4033, 4034, 5452.

[23] K. Fujiwara. Analysis of DITL root data and comparison with full-resolver's data". *DNS-OARC Spring 2014 Workshop*, May 2014.

[24] Kazunori Fujiwara. 2014 Root DITL Data analysis and TLD popularity analysis. *DNS-OARC 2014 Fall Workshop (Los Angeles)*, Oct 2014.

[25] H. Gao, V. Yegneswaran, Y. Chen, P. Porras, S. Ghosh, J. Jiang, and H. Duan. An Empirical Reexamination of Global DNS Behavior. *SIGCOMM2013 Workshop*, 2013.

[26] H. Iinou, M Zushi, H. Nishida, and K. Sato. An analysis of DNS queries sent from hosts to caching servers. *DNS-OARC Fall 2010 Workshop*, Oct 2010.

[27] Inc Internet Systems Consortium. DNS ソフトウェア BIND 9. ftp://ftp.isc.org/isc/bind9/.

[28] Tomohiro Ishihara, Hajime Tazaki, Kazuya Okada, Daisuke Miyamoto, and Yuji Sekiya. Hadoop を利用した dns トラヒックのセキュリティ解析をおこなう基盤についての一検討. 電子情報通信学会技術研究報告, Mar 2014.

[29] J. Jiang, K. Li, J. Li, H. Duan, and J. Wu. Ghost Domain Names: Revoked Yet Still Resolvable. *NDSS Symposium*, Feb 2012.

[30] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. *Networking, IEEE/ACM Transactions on*, 10(5):589–603, 2002.

[31] D. Kaminsky. Catching up with Kaminsky. *Network Security*, 2008(9):4–7, Sep 2008.

[32] Yuko Kamiya, Fuminori Tanizaki, and Toshihiko Shimokawa. Dns キャッシュのクライアントへの影響に関する調査. 電子情報通信学会技術研究報告, pages 33–37, Dec 2009.

[33] Cho Kenjiro. ブロードバンドトラフィックレポート: P2p ファイル共有から web サービスへシフト傾向にあるトラフィック. *Internet Infrastructure Review*, 8, Aug 2010.

[34] Cho Kenjiro. ブロードバンドトラフィックレポート: この 1 年でトラフィック量は着実に増加、https の利用が拡大. *Internet Infrastructure Review*, 24, Aug 2014.

[35] W. Kumari and P. Hoffman. Decreasing Access Time to Root Servers by Running One on Loopback. Nov. 2014. draft-ietf-dnsop-root-loopback-00.txt.

[36] W. Kumari and P. Hoffman. Securely Distributing the DNS Root. May 2014. draft-wkumari-dnsop-dist-root-00.txt.

[37] NLnet Labs. フルリゾルバソフトウェア Unbound. http://www.unbound.net.

[38] M. Larson and P. Barber. Observed DNS Resolution Misbehavior. RFC 4697 (Best Current Practice), October 2006.

[39] B. Laurie, G. Sisson, R. Arends, and D. Blacka. DNS Security (DNSSEC) Hashed Authenticated Denial of Existence. RFC 5155 (Proposed Standard), March 2008. Updated by RFCs 6840, 6944.

[40] T. Lee, B. Huffaker, M. Fomenkov, and k. claffy. On the problem of optimization of DNS root servers' placement. *Passive and Active Network Measurement Workshop (PAM)*, Apr 2003.

[41] W. Lian, E. Rescorla, H. Shacham, and S. Savage. Measuring the practical impact of dnssec deployment. *Proceedings of USENIX Security*, Aug 2013.

[42] C. Liu and Albitz P. *DNS and Bind*. O'Reilly Media, 2006.

[43] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. Claffy. Two days in the life of the DNS anycast root servers. *Passive and Active Network Measurement*, pages 125–134, 2007.

[44] P.V. Mockapetris. Domain names: Concepts and facilities. RFC 882, November 1983. Obsoleted by RFCs 1034, 1035, updated by RFC 973.

[45] P.V. Mockapetris. Domain names: Implementation specification. RFC 883, November 1983. Obsoleted by RFCs 1034, 1035, updated by RFC 973.

[46] P.V. Mockapetris. Domain names - concepts and facilities. RFC 1034 (INTERNET STANDARD), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936.

[47] P.V. Mockapetris. Domain names - implementation and specification. RFC 1035 (INTERNET STANDARD), November 1987. Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2673, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, 6604.

[48] P.V. Mockapetris and K.J. Dunlap. Development of the domain name system. *ACM SIGCOMM Computer Communication Review*, 25(1):112–122, 1995.

[49] B. Müller. IMPROVED DNS SPOOFING USING NODE RE DELEGATION. 2008. https://www.sec-consult.com/fxdata/ seccons/prod/downloads/ whitepaper-dns-node-redelegation.pdf.

[50] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), February 1996. Updated by RFC 6761.

[51] Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, and Aditya Yadav. A global name service for a highly mobile internetwork. *Proceedings of the 2014 ACM Conference on SIGCOMM*, pages 247–258, 2014.

[52] Sam Trenholem. Deadwood – A tiny recursive DNS server. http://maradns.samiam.org/deadwood/.

[53] P. Vixie. Extension Mechanisms for DNS (EDNS0). RFC 2671 (Proposed Standard), August 1999. Obsoleted by RFC 6891.

[54] P. Vixie, R. Joffe, and F. Neves. Improvements to DNS Resolvers for Resiliency, Robustness, and Responsiveness. June 2010. draft-vixie-dnsext-resimprove-00.txt.

[55] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard), April 1997. Updated by RFCs 3007, 4035, 4033, 4034.

[56] Zheng Wang. Analysis of dns cache effects on query distribution. *Scientific World Journal*, 2013, Dec 2013.

[57] S. Weiler. DNSSEC Lookaside Validation (DLV). RFC 5074 (Informational), November 2007.

[58] D. Wessels, M. Fomenkov, N. Brownlee, and k. claffy. Measurements and Laboratory Simulations of the Upper DNS Hierarchy. *Passive and Active Network Measurement Workshop (PAM)*, pages 147–157, Apr 2004.

[59] Duane Wessels and Matt Larson. Analysis of query traffic to .com/.net name servers. *DNS WG, RIPE 66 meeting*, May 2013.

[60] Wouter Wijngaards. Resolver side mitigations. Aug. 2008. draft-wijngaards-dnsext-resolver-side-mitigation-00.txt.

[61] Wouter Wijngaards. Resolver side mitigations. Feb. 2009. draft-wijngaards-dnsext-resolver-side-mitigation-01.txt.

[62] A. Jamakovic Y.Koc and B. Hijsen. A Global Reference Model of the DNS. *DNS EASY 2011 Workshop*, Oct 2011.

[63] Yingdi Yu, Duane Wessels, Matt Larson, and Lixia Zhang. Authority server selection in dns caching resolvers. *ACM SIGCOMM Computer Communication Review*, 42:80–86, Apr 2012.

[64] 総務省. 我が国のインターネットにおけるトラヒックの集計･試算/2013年５月の集計結果の公表. 総務省報道資料, Aug 2013.

関連業績リスト

- 公表済み論文

  (1)Kazunori Fujiwara, Akira Sato and Kenichi Yoshida,”DNS traffic analysis — CDN and the World IPv6 Launch“, Journal of Information Processing, Vol.21, No.3, pp. 517-526, July 2013.

- 査読付き国際会議論文

  (2) Kazunori Fujiwara, Akira Sato and Kenichi Yoshida, “DNS Traffic Analysis: Issues of IPv6 and CDN”,2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet, pp. 129-137, July 2012

- 投稿中の論文

  (3) 藤原和典，佐藤聡， 吉田健一,「ルート DNS サーバへのクエリ数の削減」，電子情報通信学会論文誌 B 分冊

- その他

  (4)Kazunori Fujiwara, “Analysis of DITL root data and comparison with full-resolver’s data”, DNS-OARC Spring 2014 Workshop, May 2014.

  (5)Kazunori Fujiwara, “2014 Root DITL Data analysis and TLD popularity analysis”, DNS-OARC 2014 Fall Workshop (Los

Angeles), Oct 2014.

(6) 藤原和典, 「［招待講演］「根元」は攻略されたのか　～ DNS キャッシュポイズニング攻撃とその対策について改めて考える ～」, 電子情報通信学会技術研究報,Vol.114,No.216,Sep. 2014

(7)Kazunori Fujiwara, “Side effect of DNSSEC: an increase of DS queries”, draft-fujiwara-dnsop-ds-query-increase-02, Jan. 2014